

## Анализ симуляторов систем отказоустойчивого хранения данных

*Н.С. Могилевская, А.И. Юрасов*

*Южный федеральный университет, Ростов-на-Дону*

**Аннотация:** Распределенные системы хранения данных (РСХД) являются многопараметрическими сложно настраиваемыми системами. Отказоустойчивость и надежность хранения данных РСХД обеспечиваются набором различных методов, которые должны быть эффективны не только по отдельности, но и быть хорошо согласованы между собой внутри системы хранения. В настоящее время создаются новые методы обеспечения отказоустойчивости и надежности, которые могут быть использованы в РСХД. Для оценки эффективности применения таких методов удобно использовать программные средства, моделирующие работу РСХД. Цель данной работы заключается в изучении существующих программных симуляторов РСХД для оценки потенциала использования их в моделировании отказоустойчивости и надежности хранения данных. Проведен сравнительный анализ нескольких симуляторов, которые имеют широкий набор реализуемых функций и описаны в рецензируемых публикациях. На основе анализа построена обобщенная структурная схема симулятора РСХД и сделан вывод о необходимости разработки нового симулятора в виде программного средства с открытым исходным кодом, архитектура которого рассчитана на его расширение в том числе сторонними разработчиками. Такой симулятор позволит тестировать новые разработки в области технологий повышения отказоустойчивости. В работе сформирован ряд базовых требований к реализации и функционалу, сформулированы основные рассчитываемые метрики.

**Ключевые слова:** системы хранения данных, отказоустойчивость, коды стирания, программный симулятор, надежность, имитационная модель.

### Введение

Надежное хранение данных больших объемов – это одно из важнейших требований к современным информационным системам. Объем данных, производимых человечеством, стремительно растет, и большая часть этих данных хранится не на устройствах пользователя, а в “облаке”, представляющем собой распределенное хранилище.

Распределенная система хранения данных (РСХД) представляет собой комплекс программно-аппаратных средств, обеспечивающий отказоустойчивое хранение данных большого объема и управление ими. Данные в РСХД сохраняются на большом количестве физических устройств, возможно территориально удаленных друг от друга. Одним из важных требований к РСХД является обеспечение отказоустойчивости, т.е.

---

сохранение работоспособности РСХД и целостности хранимых данных при разнообразных отказах оборудования. Для ее обеспечения в РСХД применяют комплекс различных техник [1], самые популярные из них это методы избыточного алгебраического кодирования, методы размещения блоков данных между дисками, а также методы эффективного восстановления данных после отказов оборудования.

Помехоустойчивые коды, традиционно используемые в каналах передачи данных для борьбы с ошибками, в условиях РСХД используются в режиме исправления стираний. В РСХД часто используются коды повторения (репликация данных), например, в таких системах как Windows Azure Storage (WAS), Hadoop Distributed File System (HDFS), Google File Storage (GFS) и RAMCloud [2]. Среди других алгебраических кодов, корректирующих стирания, можно отметить (10,6)-код Рида-Соломона активно использующийся в RAID-6, Quick File System (QFS) и Facebook (компания Meta признана экстремистской организацией и запрещена на территории РФ) [2], а также Local Repairable код (LRC), применяемый, например, в WAS [3].

В качестве примеров методов размещения данных между устройствами хранения назовем Spread Placement Scheme (SSS) и Partitioned Placement Scheme (PSS). Первый из них используется системами QFS и RAMCloud [1]. PSS-подход применяется в Facebook [1]. Для размещения данных между стойками в Facebook применяется Flat-подход [4]. Известными методами восстановления данных после отказа являются, например, Eager-подход, LAZY [5] и RAFI [6].

РСХД является многопараметрической сложно настраиваемой системой, в которой применяется несколько требующих слаженной работы методов. В настоящее время разрабатывается большое количество новых методов обеспечения отказоустойчивости РСХД, таких, как новые коды

---

стирания [7] и [8], а также существуют решения, которые применяются в других практических приложениях и которые потенциально могут быть применены в РСХД, например, пороговые методы разделения данных [9], эффективность которых продолжает совершенствоваться [10]. В качестве новых методов кодирования потенциально подходящими для внедрения в РСХД выглядят, например, актуальные разработки в области кодов стирания, такие как MSR-коды с малым уровнем разделения данных на блоки [11] и Clay-коды [12]. Для оценки возможности применения новых методов обеспечения отказоустойчивости целесообразно использовать симуляторы РСХД и модельные эксперименты.

Цель данной работы заключается в поиске и исследовании доступных программных симуляторов РСХД на предмет возможности тестирования с их помощью новых методов и технологий обеспечения отказоустойчивости и надежности хранения данных. Ниже в разделе *Схема имитационной модели РСХД* построена обобщенная структурная схема симуляторов РСХД. Сравнительный анализ симуляторов в рамках элементов структурной схемы приведен в разделе *Сравнительный анализ симуляторов*. Раздел *Обсуждение и заключение* посвящен оценке результатов анализа.

### **Схема имитационной модели РСХД**

При проведении исследования рассмотрен ряд разнообразных программных симуляторов. В работе ниже упоминаются только 6 из них: Comprehensive Reliability SIMulator (CR-SIM) [1], Simulator of Erasure-Coded Data Centers (SimEDC) [13], Distributed Storage Simulator (DS-SIM) [5], Cloud Quality-of-Service Simulator for Reliability (CQSim-R) [14], DR-SIM [6] и HFRS [15]. Эти симуляторы были выбраны по двум критериям, а именно, обсуждение симулятора в нескольких рецензируемых публикациях, а также широкий набор реализуемых функций. Изучаемые в работе симуляторы РСХД обладают разнообразными свойствами, оценивают множество

---

параметров, используют различные схемы повышения надежности и т.д. В процессе их анализа были выделены общие элементы архитектуры симуляторов, определены связи между элементами, и построена обобщенная структурная схема имитационной модели РСХД (рис. 1). Сравнение симуляторов производилось в соответствии с блоками построенной схемы. Опишем ее.

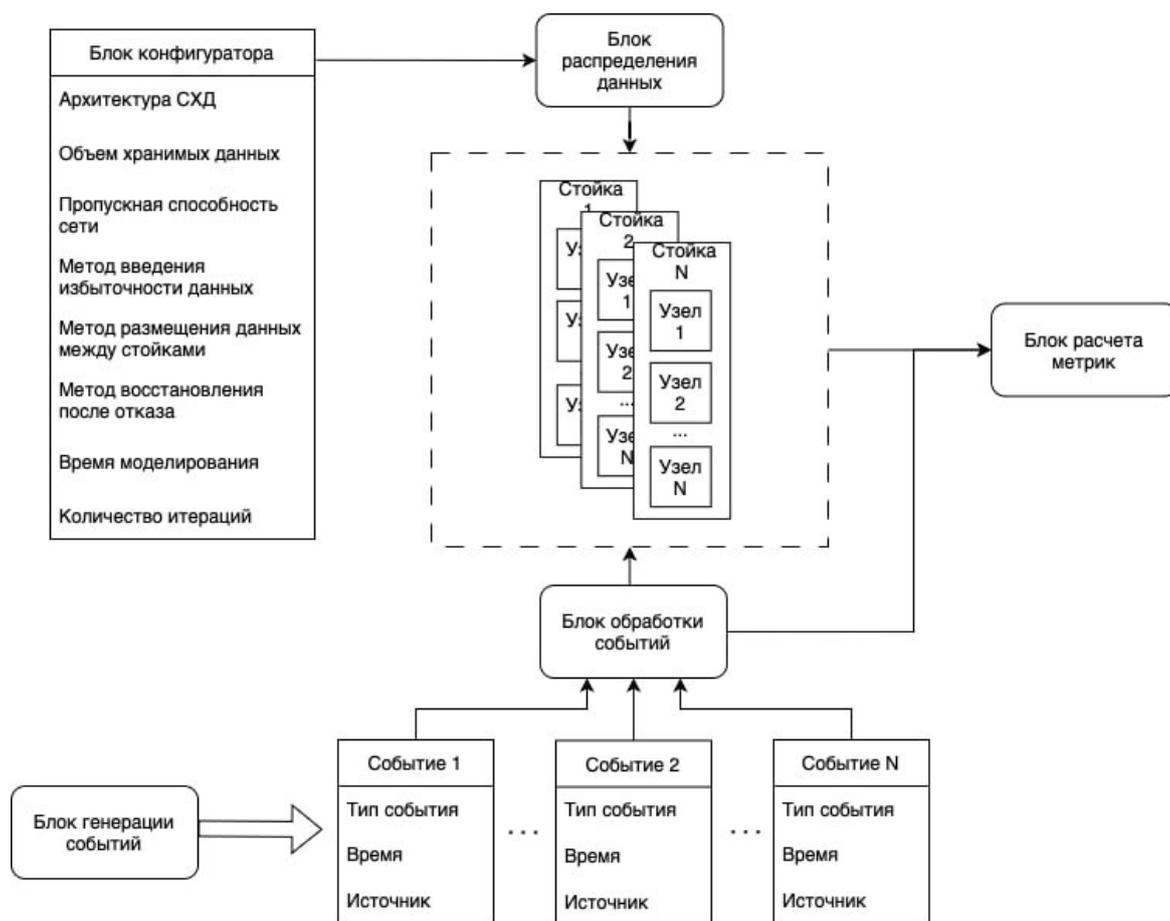


Рис. 1. - Обобщенная структурная схема имитационной модели РСХД

Настройки имитационной модели задаются в блоке-конфигураторе. В нем указывается набор параметров, который полностью определяет архитектуру и параметры РСХД, например, число дисков в массивах, общий объем хранимых данных, распределение дисков по узлам и стойкам, размеры блоков и чанков, параметры пропускной способности сети, используемая схема введения избыточности и схема размещения блоков. На основе

заданных параметров выполняется конструирование древовидной структуры из дисков, узлов и стоек. На рисунке эта структура помещена в прямоугольник, обозначенный пунктирной линией.

Блок распределения данных отвечает за моделирование отказоустойчивого хранения, а именно выполняет избыточное кодирование тестовых данных и распределение фрагментов или копий по дискам согласно выбранной схеме размещения.

Блок генерации событий формирует по заданным правилам отказы оборудования и указания по восстановлению данных. К отказам оборудования относятся отказ сектора диска, временный или постоянный отказ диска, а также временные отказы узлов и стоек. Указания по восстановлению данных формируются в момент обнаружения системой хранения выхода из строя оборудования. В симуляторе они обычно генерируются с учетом времени на идентификацию и среднему времени на восстановление оборудования.

Блок обработки событий производит разбор очереди событий от блока генерации событий, а также формирует реакцию системы хранения на очередное событие из очереди. Например, после обнаружения отказавшего элемента система должна попытаться восстановить потерянные данные по имеющимся в доступе фрагментам и переместить их в новое устройство хранения. Каждое событие вносит изменение в состояние системы, влияя на показатели отказоустойчивости, фиксируемые данным блоком.

Блок расчета метрик на основе результатов обработки событий и состояния данных в системе хранения формирует значения показателей качества системы хранения. Совокупность значений таких показателей отражает текущее состояние системы в контексте отказоустойчивости. Отметим, что в настоящее время не существует общепризнанных метрик оценки надежности и эффективности РСХД. Для оценки качества РСХД

---

используют различные вероятностные характеристики, например, такие как вероятность потери данных, среднее время до возникновения отказа или среднее время до восстановления.

### Сравнительный анализ симуляторов

Сравним симуляторы РСХД по блокам имитационной схемы. В компактном виде параметры симуляторов представлены в таблице №1.

Таблица № 1

#### Сравнительный анализ симуляторов

№ п/п	Математические модели событий отказов	Поддерживаемые схемы отказоустойчивости	Выходные метрики
CR-SIM	$W(k, \lambda)$ , $\text{Exp}(\lambda)$ , трассировки реальных РСХД	Репликация, RS, LRC и MSR	PDL, TRS, TSC
SimEDC	$\text{Exp}(\lambda)$ , трассировки реальных РСХД	RS, LRC и DRC	PDL, NOMDL, BR
DS-SIM	$\text{Exp}(\lambda)$ , трассировки реальных РСХД	Репликация, RS, Xorbas, LRC	PDL, Normalised network bandwidth
CQSim-R	$\text{Exp}(\lambda)$	Репликация, RS(6, 3)	MTTDL, MTBLE
DR-SIM	$\text{Exp}(\lambda)$	RS, Zigzag и LRC	RNT, MTTDL
HFRS	$W(k, \lambda)$	Коды FLAT, Weaver, EVENODD, SPC	MTTDL

*Математические модели потока событий.* Для организации работы блока генерации событий возможно использовать записи об отказах из логов реальных РСХД. При отсутствии или недоступности таких знаний или их малом количестве для генерации событий удобно использовать соответствующие математические модели.

Среди рассмотренных симуляторов позволяют подавать на вход трассировки событий, снятых с реальных РСХД, симуляторы DS-SIM и

SimEDC. В симуляторах CR-SIM, SimEDC, HFRS и DS-SIM события восстановления и отказа оборудования генерируются на основе экспоненциального распределения (в табл. 1 обозначено  $\text{Exp}(1)$ ) или распределения Вейбулла (в табл. 1 обозначено  $W(k, \lambda)$ ). CQSim-R, HFRS и DR-SIM используют в своей работе методы Монте-Карло для анализа случайных событий отказов.

Рассмотренные симуляторы различаются по видам моделируемых отказов. Самый широкий набор отказов предоставляют CR-SIM [1] и DS-SIM [5], они моделируют события отказа секторов диска, самих дисков, постоянные и временные отказы узлов и временные отказы стоек. SimEDC не рассматривает отказы в секторе, поддерживая все остальные, перечисленные выше типы отказов, а также предоставляет возможность симуляции множественных одновременно проявляющихся отказов.

*Поддерживаемые схемы отказоустойчивости.* Блок распределения данных отвечает за организацию отказоустойчивости системы хранения. Наиболее широким набором поддерживаемых методов повышения отказоустойчивости обладает симулятор CR-SIM. В качестве метода введения избыточности в нем используется репликация, коды Рида-Соломона (PC), а также LR и MSR коды. Для размещения данных между узлами можно использовать методы SSS, PSS, CopySet [1]. Размещение данных между стойками задается методами Flat или Hier. Метод восстановления данных представлен такими технологиями, как Eager Repair Scheme (Eager), Lazy Repair Scheme (Lazy), Risk Awareness Failure Identification Repair Scheme (RAFI) и комбинация Lazy и RAFI.

Остальные симуляторы имеют более скромные возможности и ориентированы на тестирование комбинации метода введения избыточности с какой-нибудь одной другой технологией повышения надежности.

---

Так, например, SimEDC предоставляет возможность выбрать между кодами PC, LRC и DRC для введения избыточности и один из методов размещения данных между узлами: Hier или Flat. CQSim-R позволяет моделировать PCXD с методами SSS, PSS, CopySet размещения данных, применяемых к дискам. В качестве метода избыточного кодирования CQSim-R использует коды повторения (репликацию) и код PC с фиксированными параметрами.

DS-SIM и DR-SIM моделируют PCXD с возможностью настройки избыточного кодирования и метода восстановления данных. DR-SIM поддерживает коды PC, Zigzag, LRC, а также методы RAFI и Lazy для восстановления данных. DS-SIM способен кодировать данные кодами повторения, PC, Xorbas и LRC-кодами, при этом для восстановления данных возможно использовать схему Lazy. Система HFRS поддерживает только один метод повышения отказоустойчивости, а именно избыточное кодирование. В системе доступны коды стирания FLAT, Weaver, EVENODD [15].

*Выходные метрики симулятора.* При оценке симулятора помимо отказоустойчивости важно уметь определять стоимостные характеристики работы СХД, позволяющие понять затраты на хранение или восстановление данных. Единственным из представленных симуляторов, поддерживающим оценку как отказоустойчивости, так и стоимости хранения, является CR-SIM. Он позволяет формировать в результате своей работы показатели общей стоимости хранения в ПиБ/месяц (Total storage cost - TSC) и общей стоимости восстановления в ПиБ (Total repair cost - TRC).

Симуляторы CR-SIM, SimEDC, а также DS-SIM реализуют подсчет страйпов с утерянными блоками данных, что на выходе системы формирует показатель вероятности потери данных (Probability of data loss - PDL). Эти же данные SimEDC использует для подсчета других метрик - ожидаемого

---

объема потерянных данных в байтах, нормализованного по объему хранилища (Normalized magnitude of data loss - NOMDL), и коэффициента блокировки (Blocked ratio - BR), показывающего долю времени, когда блок данных недоступен из-за отказа диска, узла или стойки, где он хранится.

Анализ потребляемого на восстановление данных трафика в системе хранения оценивается только в DS-SIM и DR-SIM как пропускная способность сети в Тб/день и нормализованный сетевой трафик восстановления данных (Repair network traffic - RNT) соответственно. Симуляторы CQSim-R, HFRS и DR-SIM позволяют также оперировать усредненными показателями. Первый из них формирует в результате своей работы две метрики - среднюю скорость потери данных, выраженную в доле потерянных блоков от общего числа в единицу времени (Mean loss rate - MLR), а также среднее время между событиями потери данных в днях (Mean time between loss event - MTBLE). DR-SIM и HFRS и генерирует похожий оценочный показатель - среднее время до потери данных (Meta time to data loss).

Отметим, что в работе не производилось сравнение симуляторов по некоторым из структурных элементов. Из-за закрытости систем и малой освещенности в статьях не представляется возможным проанализировать различие симуляторов в блоке обработки событий. По блоку конфигурирования сравнение не производилось по причине схожести их возможностей.

Виды отказов оборудования, поддерживаемые симуляторами отражены в таблице 2. В таблице использованы обозначения врем. – временный, пост. – постоянный, S – общее число поддерживаемых отказов.

Важной характеристикой симулятора является его доступность, т.е. принципиальная возможность воспользоваться программным средством. Из всех представленных симуляторов полноценно и бесплатно доступны симуляторы SimEDC и DR-SIM.

---

Таблица № 2

Моделируемые виды отказов

	Отказ сектора		Отказ диска		Отказ узла		Врем. отказа стойки	Множественный отказ	S
	врем.	пост.	врем.	пост.	врем.	пост.			
CR-SIM	+	-	+	+	+	+	+	-	6
SimEDC	-	-	-	+	+	-	+	+	4
DS-SIM	-	-	+	+	+	+	+	-	5
CQSim-R	-	-	+	+	-	-	-	-	2
DR-SIM	-	-	-	-	+	+	-	-	2
HFRS	+	+	+	+	-	-	-	-	4

Исходный код этих проектов можно найти на гитхаб – по URL: [github.com/millyz/SimEDC](https://github.com/millyz/SimEDC) для SimEDC и URL: [github.com/decoqt/distributed\\_system\\_simulator](https://github.com/decoqt/distributed_system_simulator) для DR-SIM соответственно. Страница проекта DS-SIM URL: [code.google.com/archive/p/ds-sim/](https://code.google.com/archive/p/ds-sim/) доступна в формате Google-архива, но без доступа к программному продукту. CR-SIM и CQSim-R не имеют публичной общедоступной версии. Заметим, что CR-SIM и SimEDC основаны на базе HFRS-симулятора [15], аналог которого можно

найти в открытом доступе – URL: [github.com/fomy/simd](https://github.com/fomy/simd). Сведения о доступности симуляторов приведены в таблице № 3.

Таблица № 3

Доступность симуляторов

	Доступ к ПС	Доступ к исходным кодам
CR-SIM	-	-
SimEDC	+	+
DS-SIM	-	-
CQSim-R	-	-
DR-SIM	+	+
HFRS	-	+

**Обсуждение и заключение**

Проведенный сравнительный анализ свойств симуляторов РСХД показал, что наиболее широким функционалом из представленных моделей обладает CR-SIM симулятор. Согласно [1], он поддерживает работу различных методов повышения отказоустойчивости, поддерживает симуляцию наибольшего количества типов отказов, а также обладает гибкой архитектурой, позволяющей проводить точную настройку различных параметров РСХД. Это единственный симулятор из рассмотренных, использующий показатели стоимости для операций по хранению и восстановлению данных. Однако невозможность получения доступа к программному продукту и его исходным кодам является недостатком, который перечеркивает все достоинства CR-SIM.

Симуляторы SimEDC, DS-SIM и DR-SIM имеют более узкий спектр предлагаемых методов повышения отказоустойчивости и больше направлены на проверку некоторых гипотез, выдвигаемых их авторами. Так, например, SimEDC предназначен в основном для оценки влияния выбора схемы размещения данных между узлами на параметры системы, DS-SIM призван определить влияние схемы восстановления LAZY [5] на отказоустойчивость,

а основной целью DR-SIM является подтверждение актуальности использования схемы восстановления данных RAFI [6]. Несмотря на такую узконаправленность в сравнении с CR-SIM, упомянутые симуляторы в качестве выходных данных формируют оценку разных, не встречающихся в других системах, параметров, таких как: пропускная способность сети, доля блоков данных, отказавших в доступе вследствие действия процедуры восстановления.

Симулятор CQSim-R отличается заметно упрощенной моделью отказов - поддерживаются только отказы дисков, а также симулятор поддерживает небольшой набор методов введения избыточности данных. Однако, важным преимуществом по сравнению с другими программными средствами является формирование усредненных метрик на выходе системы, наиболее привычных для оценки отказоустойчивости РСХД: среднее время между событиями потери данных и средняя скорость потери данных.

HFRS позволяет оперировать только дисковыми массивами и слабо отражает реальные процессы, происходящие в распределенных системах хранения. Однако, необходимо отметить, что архитектурные принципы его построения послужили основой для других моделей [15].

Исследование и сравнительный анализ симуляторов РСХД показал, что программный код находится в открытом доступе только у экземпляров, рассчитанных на решение узконаправленных задач. Однако, такие симуляторы тяжело модернизировать, так как разработчики этих программных продуктов изначально не проектировали их архитектуру как гибкую и расширяемую. Это накладывает серьезные ограничения на дальнейшее увеличение набора поддерживаемых ими функций. Такой подход не позволяет моделировать работу новых технологий и методов повышения отказоустойчивости.

---

Еще одним недостатком рассмотренных экземпляров симуляторов можно считать отсутствие отдельного модуля, отвечающего за симуляцию потока событий чтения и записи данных. Такие события нагружают РСХД, а рост трафика между элементами системы, в свою очередь, негативно сказывается на времени работы алгоритмов декодирования данных и восстановления утерянных фрагментов, и, следовательно, оказывает влияние на общие показатели отказоустойчивости системы.

Основываясь на результатах сравнительного анализа симуляторов РСХД, считаем целесообразным разрабатывать открытую, свободно-распространяемую программу-симулятор систем отказоустойчивого хранения данных, архитектура которой предполагает расширение, изменение и улучшение симулятора в том числе и сторонними разработчиками. Такой симулятор должен позволять задавать следующие параметры:

- объем хранилища данных и его архитектуру - количество дисков в узлах, узлов в стойках;
- суммарное время хранения;
- пропускную способность сети связи между частями системы;
- технологию размещения блоков данных между узлами и/или стойками;
- технологию введения избыточности данных.

В результате моделирования работы систем хранения данных разработчикам требуется получить ряд показателей, определяющих соответствие системы с точки зрения архитектуры, технологий обеспечения отказоустойчивости и их параметров, предъявляемым требованиям к хранению данных. Полагаем, что симулятор должен вычислять следующие параметры системы, но не ограничиваться ими:

- совокупная стоимость хранения данных, рассчитываемая из объема и времени хранения;
-

- вероятность потери данных для оценки отказоустойчивости системы;
- средняя пропускная способность сети для учета влияния алгоритмов восстановления в случае отказа;

- суммарное время недоступности при запросе различных блоков данных.

Проведенный анализ показал, что описанные выше метрики, представляют интерес для разработчиков РСХД и поэтому должны быть включены в создаваемый симулятор.

Разрабатывается большое количество новых технологий, которые возможно могли бы быть успешно применены в системах хранения данных большого объема. Однако перед принятием решения о внедрении новых методов в реальные РСХД необходимо провести качественные имитационные эксперименты на моделях систем хранения данных. В работе изучен ряд программных симуляторов, реализующих имитационные модели распределенных систем хранения. По результатам анализа симуляторов РСХД среди рассмотренных программных средств не найдено симуляторов, которые можно было бы расширить и использовать для исследования применимости новых технологий в РСХД. Основными причинами этого являются либо недоступность симуляторов и их исходных кодов, либо закрытость архитектуры симулятора, не предполагающая его легкое расширение и модификацию. Отмечено также, что ни один из симуляторов не содержит в своей структуре блока имитации нагрузки на чтение/запись, без которого модель РСХД представляется неполной.

### Литература

1. Qi Y., Feng D. and Hou B. Towards building reliable and cost-efficient distributed storage systems. IEEE Access. 2020. 8. Pp. 157862-157877.

2. Saadoon M., Hamid S.H., Sofian H., Altarturi H.M. and Azizul Z.H. Fault tolerance in big data storage and processing systems: A review on challenges and solutions. *Ain Shams Engineering Journal*. 2022. 13. 2. 101538 p.
3. Zhang M., Han S., Lee P.C. SimEDC: A Simulator for the Reliability Analysis of Erasure-Coded Data Centers. *IEEE Transactions on Parallel and Distributed Systems*. 2019. 30. 12. Pp. 2836–2848.
4. Rashmi K., Shah N.B., Gu D., Kuang H., Borthakur D. A Hitchhiker’s Guide to Fast and Efficient Data Reconstruction in Erasure-coded Data Centers. *Proc. of ACM SIGCOMM*. New York. ACM. 2014. Pp. 331–342.
5. Silberstein M., Ganesh L., Wang Y., Alvisi L., Dahlin M. Lazy means smart: Reducing repair bandwidth costs in erasure-coded distributed storage. *Proceedings of International Conference on Systems and Storage*. New York. ACM. 2014. Pp. 1–7.
6. Fang J., Wan S., He X. RAFI: Risk-aware failure identification to improve the RAS in erasure-coded data centers. *Proceedings of USENIX Annual Technical Conference (USENIX ATC '18)*. Berkeley. Usenix. 2018. Pp. 495-506.
7. Shi H., Hou H. New piggybacking codes with lower repair bandwidth for any single-node failure. *Proceedings of IEEE International Symposium on Information Theory (ISIT)*. Espoo. IEEE. 2022. Pp. 2601–2606.
8. Silberstein N., Etzion T. Optimal fractional repetition codes based on graphs and designs. *IEEE Transactions on Information Theory*. 2015. 61. 8. Pp. 4164–4180.
9. Могилевская Н.С. Дифференцирование полиномов нескольких переменных над полями Галуа и приложения к кодам Рида–Маллера. Ростов-на-Дону – Таганрог. Южный федеральный университет. 2022. 124 с.

10. Mogilevskaya N.S. Data sharing method with error protection of shares. Proceedings of the 17th IEEE International Conference on Application of Information and Communication Technologies (AICT 2023). Baku. IEEE. 2023.
11. Rawat A.S., Tamo I., Guruswami V., Efremenko K.  $\epsilon$ -MSR codes with small sub-packetization. Proceedings of IEEE International Symposium on Information Theory. Aachen. IEEE. 2017. Pp. 2043–2047.
12. Vajha M., Ramkumar V., Puranik B., Kini G., Lobo E., Sasidharan B. et al. Clay codes: moulding MDS codes to yield an MSR code. Proceedings of the 16th USENIX Conference on File and Storage Technologies. Berkeley. USENIX. 2018. Pp. 139-153.
13. Huang C., Simitci H., Xu Y., Ogun A., Calder B., Gopalan P. et al. Erasure coding in Windows Azure Storage. Proceedings of the USENIX Conference on Annual Technical. Boston. USENIX. 2012. Pp. 15–26.
14. Hall R.J. Tools for predicting the reliability of large-scale storage systems. ACM Transactions on Storage. 2016. 12. 4. Pp. 1–30.
15. Greenan K.M. Reliability and power-efficiency in erasure-coded storage systems. Santa Cruz. UCSC. 2009. 187 p.

### References

1. Qi Y., Feng D. and Hou B. Towards building reliable and cost-efficient distributed storage systems. IEEE Access. 2020. 8. Pp. 157862-157877.
  2. Saadoon M., Hamid S.H., Sofian H., Altarturi H.M. and Azizul Z.H. Fault tolerance in big data storage and processing systems: A review on challenges and solutions. Ain Shams Engineering Journal. 2022. 13. 2. 101538 p.
  3. Zhang M., Han S., Lee P.C. SimEDC: A Simulator for the Reliability Analysis of Erasure-Coded Data Centers. IEEE Transactions on Parallel and Distributed Systems. 2019. 30. 12. Pp. 2836–2848.
-

4. Rashmi K., Shah N.B., Gu D., Kuang H., Borthakur D. A Hitchhiker's Guide to Fast and Efficient Data Reconstruction in Erasure-coded Data Centers. Proc. of ACM SIGCOMM. New York. ACM. 2014. Pp. 331–342.
5. Silberstein M., Ganesh L., Wang Y., Alvisi L., Dahlin M. Lazy means smart: Reducing repair bandwidth costs in erasure-coded distributed storage. Proceedings of International Conference on Systems and Storage. New York. ACM. 2014. Pp. 1–7.
6. Fang J., Wan S., He X. RAFI: Risk-aware failure identification to improve the RAS in erasure-coded data centers. Proceedings of USENIX Annual Technical Conference (USENIX ATC '18). Berkeley. Usenix. 2018. Pp. 495-506.
7. Shi H., Hou H. New piggybacking codes with lower repair bandwidth for any single-node failure. Proceedings of IEEE International Symposium on Information Theory (ISIT). Espoo. IEEE. 2022. Pp. 2601–2606.
8. Silberstein N., Etzion T. Optimal fractional repetition codes based on graphs and designs. IEEE Transactions on Information Theory. 2015. 61. 8. Pp. 4164–4180.
9. Mogilevskaya N.S. Differentirovaniye polinomov neskol'kikh peremennykh nad polyami galua i prilozheniya k kodam ridA–mallera. [Differentiation of polynomials of several variables over Galois fields and applications to Reed–Muller codes] Rostov-na-Donu – Taganrog. Yuzhnyj federal'nyj universitet. 2022. 124 p.
10. Mogilevskaya N.S. Data sharing method with error protection of shares. Proceedings of the 17th IEEE International Conference on Application of Information and Communication Technologies (AICT 2023). Baku. IEEE. 2023.



11. Rawat A.S., Tamo I., Guruswami V., Efremenko K.  $\epsilon$ -MSR codes with small sub-packetization. Proceedings of IEEE International Symposium on Information Theory. Aachen. IEEE. 2017. Pp. 2043–2047.
12. Vajha M., Ramkumar V., Puranik B., Kini G., Lobo E., Sasidharan B. et al. Clay codes: moulding MDS codes to yield an MSR code. Proceedings of the 16th USENIX Conference on File and Storage Technologies. Berkeley. USENIX. 2018. Pp. 139-153.
13. Huang C., Simitci H., Xu Y., Ogus A., Calder B., Gopalan P. et al. Erasure coding in Windows Azure Storage. Proceedings of the USENIX Conference on Annual Technical. Boston. USENIX. 2012. Pp. 15–26.
14. Hall R.J. Tools for predicting the reliability of large-scale storage systems. ACM Transactions on Storage. 2016. 12. 4. Pp. 1–30.
15. Greenan K.M. Reliability and power-efficiency in erasure-coded storage systems. Santa Cruz. UCSC. 2009. 187 p.

**Дата поступления: 14.12.2024**

**Дата публикации: 29.01.2025**