УДК 004.62, 004.7
doi:10.31799/1684-8853-2026-5-73-84
EDN: CWRBDL

Научные статьи
Articles

Средний возраст информации в распределенных системах хранения данных с равноправными узлами

Д. Р. Крылов[a], *аспирант, orcid.org/ 0009-0008-5901-5342*
У. А. Болотских[a], *магистрант, orcid.org/0009-0009-0733-594X*
А. М. Тюрликов[a], *доктор техн. наук, профессор, orcid.org/0000-0001-7132-094X, turlikov@vu.spb.ru*
[a]*Санкт-Петербургский государственный университет аэрокосмического приборостроения, Б. Морская ул., 67, Санкт-Петербург, 190000, РФ*

Введение: *распределенные системы хранения данных могут быть реализованы с помощью ведущих и ведомых узлов или на основе равноправных узлов. Для систем на основе равноправных узлов влияние параметров системы на такой показатель качества функционирования, как средний возраст информации, малоизучен.* ***Цель:*** *разработать модель реплицируемой системы хранения данных с равноправными узлами, позволяющую сформулировать и решить задачу минимизации среднего возраста информации за счет выбора размера кворума записи.* ***Результаты:*** *предложена модель, отражающая основные особенности функционирования реплицируемых систем хранения данных с равноправными узлами, на которые в многоадресном режиме по ненадежному каналу передается обновление. События, что обновление успешно доставлено, происходят независимо для разных узлов. Вероятность успешной доставки обновления является одним из параметров модели. Указан способ вычисления верхней оценки для среднего возраста информации. Приведено сравнение с имитационным моделированием и установлено, что эта оценка отличается от точного значения не более чем на 1 %. Сформулирована и решена оптимизационная задача выбора размера кворума записи, минимизирующего средний возраст информации. Численные результаты показали, что оптимальный размер кворума записи растет с увеличением общего числа узлов. Получена зависимость среднего возраста информации от общего числа узлов при оптимальном размере кворума записи и фиксированных других параметров модели. Отмечено наличие минимума в этой зависимости.* ***Практическая значимость:*** *предложенная модель отражает общие особенности реплицируемых систем хранения данных с равноправными узлами и может быть использована при проектировании реальных систем для соблюдения баланса согласованности и доступности.* ***Обсуждение:*** *ключевым допущением модели является независимость успешной доставки обновления на разные узлы. В реальных системах при многоадресной передаче это допущение может не выполняться, что требует дальнейшего исследования и учета зависимости между событиями доставки.*

Ключевые слова *— средний возраст информации, системы хранения данных, репликация, кворум чтения, кворум записи.*

Для цитирования: Крылов Д. Р., Болотских У. А., Тюрликов А. М. Средний возраст информации в распределенных системах хранения данных с равноправными узлами. *Информационно-управляющие системы*, 2026, № 5, с. 73–84. doi:10.31799/1684-8853-2026-5-73-84, EDN: CWRBDL
For citation: Krylov D. R., Bolotskikh U. A., Turlikov A. M. Average age of information in distributed data storage systems with peer nodes. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2026, no. 5, pp. 73–84 (In Russian). doi:10.31799/1684-8853-2026-5-73-84, EDN: CWRBDL

Введение

Распределенные системы хранения данных (РСХД) широко используются в современных дата-центрах для работы с большими объемами информации. Для предотвращения потери данных из-за выхода из строя узлов [1] в подобных системах применяется дублирование (репликация) фрагментов информации (реплик) на нескольких узлах [2–4]. Но так как во многих сервисах данные регулярно изменяются, то необходимо поддерживать актуальность информации, хранящуюся на узлах, а значит, обновлять уже хранящуюся информацию до более новой. В связи с этим возникает задача обеспечения определенного уровня согласованности в работе системы, чтобы избежать конфликтов при выполнении параллельных операций с данными [5, 6]. Операции, работающие с одной версией данных, называются согласованными. Согласно теореме CAP (Consistency, Availability, Partition tolerance) [7–9], в РСХД нет возможности достичь одновременно высоких показателей согласованности и доступности, что связано с взаимным влиянием этих принципов друг на друга [10]. Следовательно, возникает необходимость в поиске баланса между доступностью и согласованностью.

Основными операциями в современных дата-центрах являются запись и чтение, а самая частая — перезапись уже хранящихся данных [11, 12]. В процессе обновления данных сохраняется временная метка обновления — момент, когда данные были перезаписаны.

Существуют подходы организации работы с данными в таких системах для достижения определенного уровня согласованности и доступности.

Среди них можно выделить системы, которые в англоязычной литературе называются “Leader-based systems” [13, 14]. Они содержат в себе узлы двух типов: “leader” и “follower”, — которые в силу отсутствия устоявшейся терминологии в русском языке будем называть соответственно «ведущие» и «ведомые». Суть подхода заключается в том, что из общего множества узлов выбираются ведущие узлы, на которых обновление информации происходит всегда гарантированно, а на остальных, ведомых, лишь с некоторой вероятностью, обусловленной возможными ошибками в канале и прочими ситуациями, которые могут возникать в результате работы реальных систем [14]. Такой подход позволяет достичь относительно высокого уровня согласованности при приемлемой доступности. Представителями подобных систем на практике можно назвать Amazon DynamoDB, Google Spanner и Apple Foundation, использующие алгоритмы Paxos; для альтернативного алгоритма Raft есть такие хранилища, как MongoDBm и InfluxDB [15–19].

“Leader-based systems” — это частный случай подхода на основе кворумов [20], где нет фиксированных ведущих и ведомых узлов, а обновление должно достичь некоторого минимально необходимого числа узлов (размер кворума записи), после которого процесс обновления считается успешно выполненным и инициируется переход к следующей итерации записи. В этом случае узлами, подлежащими перезаписи, могут оказаться любые из их общего множества, а сама запись происходит только с помощью многоадресной рассылки, которая не гарантирует успешной записи. Такой подход позволяет достичь высокого уровня доступности при удерживании оптимального уровня согласованности. Данный подход рассматривается в настоящей работе и в англоязычной литературе носит название “Dynamo-style data storage” [21]. Такое название не отражает особенностей подобных систем, и прямой перевод на русский язык не имеет смысла, поэтому мы данный способ построения РСХД будем называть как системы хранения данных «с равноправными узлами». Поскольку для Dynamo-style data storage существуют понятия кворума чтения и кворума записи, а для систем Leader-based существуют понятия только кворума чтения, то системы Dynamo-style также можно называть системами «на основе кворумов».

В связи с наличием временных меток обновления на узлах для отслеживания актуальности данных удобно применить метрику, заимствованную из области систем массового обслуживания, — возраст информации (Age of Information, AoI), [22, 23]. Данная метрика позволит проанализировать работу рассматриваемой далее РСХД.

Работ с исследованиями среднего возраста информации в РСХД на основе кворумов проводилось немного. Например, в работе [21] изучалось влияние размера кворума записи на средний возраст информации в системе, получаемой операцией чтения. По результатам данной работы можно судить о наличии оптимального состояния системы, при котором средний возраст информации минимизируется. В работе [24] исследования были продолжены в сторону динамической периодической синхронизации для установления оптимального периода синхронизации при переменной нагрузке.

При увеличении размера кворума записи увеличивается и время, необходимое на запись обновления, что негативно сказывается на среднем возрасте информации. Но при малом размере кворума шанс чтения актуальной версии данных операцией чтения уменьшается, что также ведет к увеличению среднего возраста информации. Таким образом, в работе системы могут существовать оптимальные наборы параметров, при которых средний возраст информации будет минимальным.

В работе [21] время считается непрерывным, а задержки записи моделируются непрерывными случайными величинами (сдвинутое экспоненциальное распределение), от чего процесс записи начинается и заканчивается в произвольные моменты времени. В настоящей статье вся работа системы организована в кадрах, каждый из которых состоит из целого числа слотов (единиц времени), что соответствует реальным системам, где операции планируются по тактам или тикам таймера. Другим важным недостатком работы [21] является то, что не учитывается вероятность отказа при передаче, потому что предполагается, что запись на узел в конечном счете всегда происходит, просто с разной задержкой, которая представляет из себя одну непрерывную случайную величину. В нашем исследовании в каждом слоте делается попытка записи на все узлы, и она успешна с вероятностью p независимо для каждого узла, что позволяет учитывать потери пакетов, тайм-ауты, сетевые ошибки и другие сбои, характерные для реальных РСХД. Еще один недостаток в работе [21] заключается в том, что задержка между завершением одной записи и началом следующей учитывается через сдвиг c в экспоненциальном распределении, но этот сдвиг не интуитивен и не отделен от процесса записи. Нами введена явная фиксированная задержка c слотов между кадрами, что соответствует реальным системам, где после завершения записи требуется время на обработку, формирование следующего обновления, синхронизацию.

В представленной работе исследуется влияние размера кворума записи на средний возраст информации, полученной запросом на чтение.

Время разделено на слоты, в каждом из которых происходит попытка записи обновления на все узлы системы. Успешная запись происходит с некоторой вероятностью, после чего данные сразу становятся доступными для чтения. Запрос на чтение отправляется на некоторое число случайно выбранных узлов, что дает результат, по которому можно судить о среднем возрасте информации.

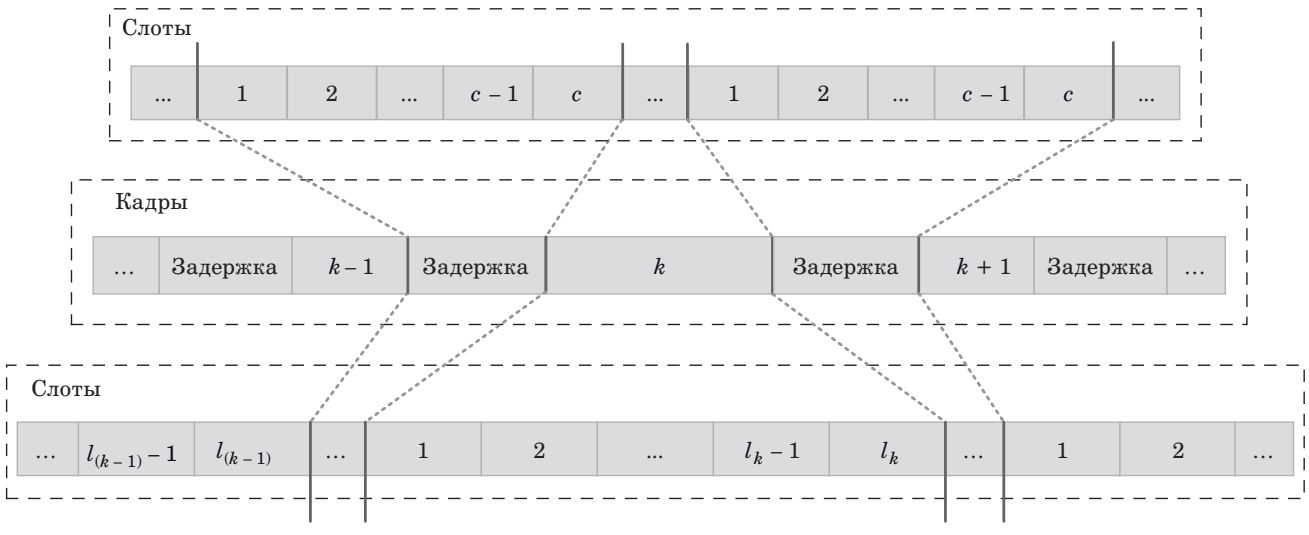
Описание модели системы

Рассмотрим систему хранения данных на основе кворумов для чтения и записи. Всего в системе содержится n узлов, все они имеют один и тот же фрагмент данных, называемый репликой. В системе существуют понятия кворума чтения R и кворума записи W. Кворум записи представляет из себя подмножество размером w узлов, а кворум чтения составляют r узлов. Таким образом, $0 < w \le n$, как и $0 < r \le n$. Узлы, находящиеся в кворуме записи, являются минимально необходимым множеством, которое будет хранить самую последнюю версию реплик. В кворуме чтения содержатся узлы, с которых происходит чтение данных.

Работа системы по времени последовательно разделяется на кадры (рис. 1), каждый из которых имеет свой порядковый номер, обозначенный k. Кадры делятся на временные интервалы, называемые слотами. За единицу времени взят один слот. Отдельно взятые кадры могут отличаться по длительности (числу слотов l_k в кадре), поскольку начало следующего кадра связано с событием инициализации нового обновления. Между кадрами имеется задержка записи, которая составляет фиксированное время, равное c слотам.

Любой кадр начинается с инициализации нового обновления, после чего в каждом слоте кадра происходит попытка записи на узлы системы. Инициализированное обновление считается самым актуальным до конца текущего кадра. Запись происходит посредством многоадресной рассылки, поэтому обновление может не дойти сразу до всех узлов-получателей, в связи с чем может потребоваться большее число слотов внутри кадра. Это обусловлено возможными ошибками в канале, потерями пакетов данных и другими причинами, следующими из реальных условий работы систем. Когда нужное количество узлов заполняет кворум записи, кадр заканчивается, и происходит задержка до следующего кадра. Фиксация успешно записанного обновления на узле происходит в конце слота, после чего оно может быть получено операцией чтения, которая может быть осуществлена в любом слоте во время задержки.

Источник осуществляет запись обновления с помощью многоадресной рассылки на все n узлов системы, но ожидает не менее w откликов с подтверждением успешной записи. Положительно ответившие узлы входят в подмножество W. Будем считать, что весь процесс записи на узел занимает один слот. Многоадресная передача не гарантирует своевременное и корректное получение информации узлом, следовательно, успешная доставка описывается вероятностью p. События успешной записи обновления реплики для разных узлов независимы. Попытка записи происходит в начале каждого слота. Процесс обновления считается завершенным после получения не менее w подтверждений от n узлов системы, при этом положительный ответ от узла засчитывается только один раз за все слоты кадра.

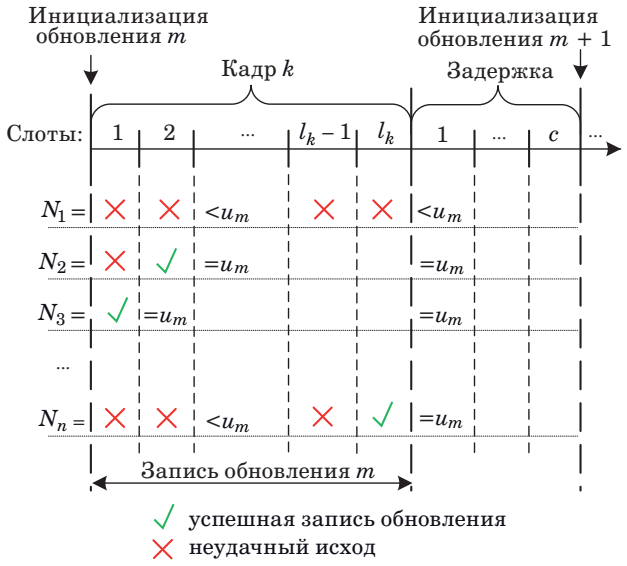


■ ***Рис. 1.*** Разделение времени работы системы на кадры и слоты
■ ***Fig. 1.*** Running time segmentation into frames and slots in the system

Таким образом, запись одного обновления может растягиваться на множество слотов, а после завершения кадра уникальных успешных записей на узлы может быть больше, чем w. После окончания текущего кадра инициализация нового обновления (появление следующего кадра) происходит только через c слотов, что обусловлено задержкой записи.

Рассмотрим процесс записи обновления более детально (рис. 2). При удачной перезаписи реплики на узле сохраняется временная метка обновления u_m (m — порядковый номер обновления), которая представляет момент, когда новая запись была инициирована. Отсчет времени начинается с запуска системы. Полная запись обновления занимает один кадр, после чего следует задержка длительностью c слотов перед началом нового кадра со следующим обновлением.

Рассмотрим процесс записи обновления с номером m в кадре k. Инициализация происходит в момент времени u_m. В каждом слоте осуществляется попытка записи обновления на все узлы системы с вероятностью p, что является показателем надежности доставки данных. При успешном получении данных временные метки узлов N_i, где $i \in \{1, 2, \ldots, n\}$ — номер узла системы, перезаписываются новой, соответствующей обновлению m, и хранятся до следующей перезаписи другим обновлением. Фиксация обновления на узле происходит сразу же в конце слота, в котором оно было получено, что дает возможность быть полученным запросом на чтение. Оно может быть произведено в любом слоте любого кадра, в том числе и в период задержки. Попытки записи предпринимаются до тех пор, пока не наберется в совокупности как минимум w узлов с последней версией данных.



■ ***Рис. 2.*** Процесс записи m-го обновления
■ ***Fig. 2.*** The process of writing the m-th update

Так как $w < n$, то к концу процесса обновления могут остаться узлы с более старой версией данных, например, как у N_1, куда обновление так и не смогло записаться (см. рис. 2). Внутри кадра находится l_k слотов, где k — номер кадра.

Получение информации с узлов осуществляется с помощью запросов на чтение, которые отправляются случайному набору узлов в количестве r, не превышающем общего количества узлов в системе. Эти r узлов образуют множество R, называемое кворумом чтения. Случайный выбор узлов отражает равномерное распределение нагрузки, что имеет место в реальных системах. По причине того, что на разных случайно выбранных узлах может содержаться информация разной свежести, среди них берется реплика с самой новой временной меткой. Чтение информации возможно только в конце слота. Операции чтения и записи производятся независимо друг от друга.

Узел, хранящий последнюю версию данных, называется согласованным. Если согласованный узел попал в кворум чтения, то и операция чтения станет согласованной, передав актуальную информацию. В кворумных системах существуют строгая и нестрогая согласованности. Нестрогая согласованность (или нестрогий кворум) описывает случай, когда кворумы записи и чтения могут не пересекаться, т. е. $w + r \le n$, и тогда гарантий получения последнего обновления операцией записи нет. Для строгой согласованности (или строгого кворума) кворумы записи и чтения должны пересекаться хотя бы в одном узле, т. е. $w + r > n$. Получение актуальных данных при этом будет гарантировано, поскольку кворумы записи и чтения будут содержать общие узлы.

Данное исследование посвящено анализу среднего возраста информации, который может быть получен с помощью запроса на чтение. Возраст информации $\Delta_i(t)$ для i-го узла в момент времени t определяется как разница между текущим временем t и меткой во времени $N_i(t)$ последнего обновления данного узла:

$$\Delta_i(t) = t - N_i(t). \tag{1}$$

Что касается запроса на чтения: возраст информации $\Delta(t)$ в момент времени t представляет собой наименьшее значение среди показателей устаревания r узлов, которые были задействованы в запросе:

$$\Delta(t) = \min(\Delta_1(t), \ldots, \Delta_r(t)). \tag{2}$$

Для вычисления среднего возраста информации $\overline{\Delta}$ применяется формула

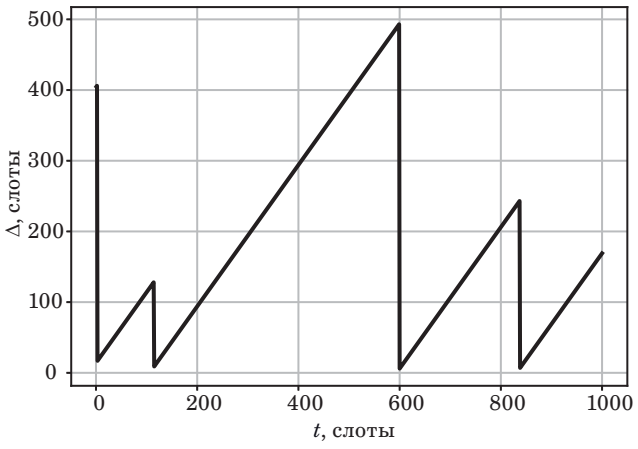
$$\overline{\Delta} = \lim_{T\to\infty} \frac{\int_0^T \Delta(t)\mathrm{d}t}{T}. \tag{3}$$

Представленная выше модель РСХД с равноправными узлами задается при помощи следующего набора переменных: n — общее число узлов в системе; w — число узлов, входящих в кворум записи; r — число узлов кворума чтения; p — вероятность успешного выполнения записи на узлы; c — число слотов задержки перед записью. Для демонстрации функционирования системы разработана имитационная модель в виде программного обеспечения.

Запросы на чтение с узла проводились в период после стабилизации системы, когда система проработала некоторое длительное время. Возраст информации $\Delta_i(t)$ рассчитывался согласно выражению (1). Возрастной процесс информации на узле (рис. 3) имеет сложный характер, следовательно, определение среднего возраста информации является отдельной задачей.

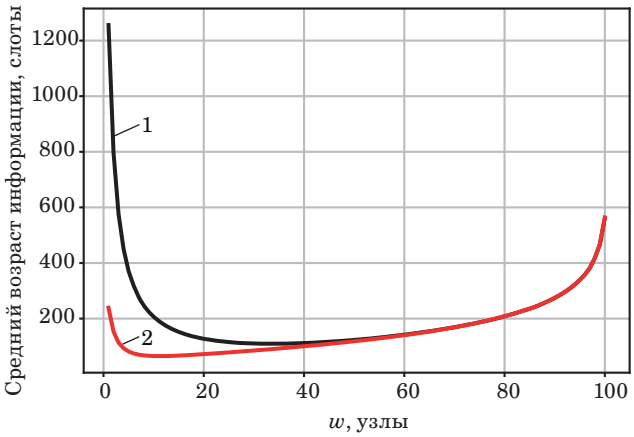
Зависимость среднего возраста информации от размера кворума записи w показана на рис. 4. На графиках определяются минимумы среднего возраста информации: для графика *1* — примерно в районе 35 узлов в кворуме записи, а для графика *2* минимум смещен и находится вблизи 11 узлов.

Средний возраст информации, получаемый из системы при большем числе узлов кворума чтения, оказался свежее. Такое поведение графиков с выраженным минимумом можно объяснить тем, что при малом числе w слишком мало узлов в конечном итоге получают обновление, из-за чего вероятность попадания таких узлов в кворум чтения R слишком низкая. При больших значениях w средний возраст информации увеличивается из-за излишне длительного процесса записи обновления, что переходит в ощутимую задержку до инициализации следующего обновления.



■ ***Рис. 3.*** Возраст информации по запросу на чтения с четвертого узла

■ ***Fig. 3.*** Age of information upon a read request from fourth node



■ ***Рис. 4.*** Зависимость среднего возраста информации от количества узлов w в кворуме записи при $r = 5$ (линия *1*) и $r = 25$ (линия *2*); $n = 100$, $p = 0{,}01$, $c = 100$

■ ***Fig. 4.*** Graph of the dependence of the average information age on the number of nodes in the write quorum w for $r = 5$ (line *1*) and $r = 25$ (line *2*); $n = 100$, $p = 0.01$, $c = 100$

Определение зависимости среднего возраста информации от параметров модели без имитационного моделирования

Утверждение 1. Длина кадра $X_{(w)}$ — это случайная величина, распределенная как w-я порядковая статистика из n случайных величин, распределенных по геометрическому закону с параметром p.

Доказательство:

Пусть появилось обновление, и нужно выполнить процесс записи. В каждую единицу времени источник рассылает обновление всем узлам, на которых еще не была выполнена успешная операция записи. Для каждого узла попытка операции записи является независимой и завершается успешно с вероятностью p. После успешной записи обновления узел больше не участвует в процессе записи. Весь цикл записи обновления продолжается до тех пор, пока не будет получено как минимум w подтверждений.

Для каждого узла $i = 1, \ldots, n$ введем случайную величину X_i — время (в единицах), которое потребовалось узлу для успешной записи. Величины X_i являются независимыми и имеют геометрическое распределение с параметром p.

Пусть $X_{(1)} \le X_{(2)} \le \ldots \le X_{(n)}$ — упорядоченные значения выборки $\{X_1, \ldots, X_n\}$. Тогда момент завершения процесса записи определяется величиной $X_{(w)}$ — w-й порядковой статистикой выборки объемом n. □

В данной работе математическое ожидание w-й порядковой статистики из n независимых геометрически распределенных случайных величин будет обозначаться как $\mu_{w:n}$, что является средней длиной кадра.

Утверждение 2. Математическое ожидание длины кадра может быть вычислено следующим образом:

$$\mu_{w:n} = \frac{(1-p)^n + \sum_{i=1}^{w-1} C_n^i p^i (1-p)^{n-i} (1+\mu_{w-i:n-i}) + \sum_{i=w}^{n} C_n^i p^i (1-p)^{n-i}}{1-(1-p)^n}, \quad (4)$$

а для первой итерации

$$\mu_{1:n} = \frac{1}{1-(1-p)^n}. \quad (5)$$

Доказательство:

Для нахождения среднего времени процесса записи нужно вычислить математическое ожидание w-й порядковой статистики из n независимых геометрически распределенных случайных величин $E[X_{(w)}]$.

В каждую единицу времени может быть три состояния:

1) не выполнена ни одна успешная операция записи. С вероятностью $C_n^0 p^0 (1-p)^n$ ни один узел не выполняет операцию записи. В этом случае затрачивается одна единица времени, и система остается в исходном состоянии;

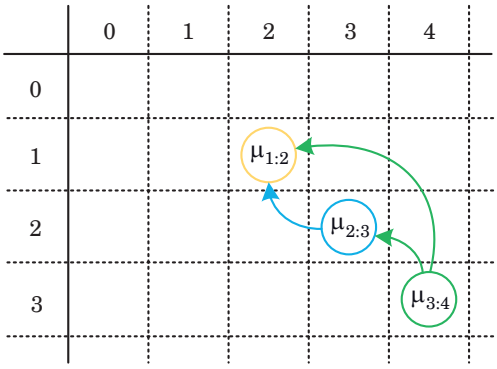
2) успешно выполнено i операций записи, где $0 < i < w$. В этом случае обновления были успешно записаны на i различных узлах с вероятностью $C_n^i p^i (1-p)^{n-i}$. Для завершения процесса записи требуется еще $w - i$ успешных операций записи по оставшимся $n - i$ узлам. Процесс продолжается с измененными параметрами;

3) успешно выполнено $i \geq w$ операций записи. Условие окончания процесса записи выполнено, цикл записи завершается, и текущее время фиксируется как случайная величина $X_{(w)}$. Событие происходит с вероятностью $C_n^i p^i (1-p)^{n-i}$.

Соответственно, для первой итерации формулы выражение примет вид

$$\mu_{1:n} = \frac{(1-p)^n + \sum_{i=1}^{n} C_n^i p^i (1-p)^{n-i}}{1-(1-p)^n} = \frac{1}{1-(1-p)^n}.$$

Таким образом, величина $X_{(w)}$ является временем достижения состояния, в котором успешно выполнено w и больше операций записи. □



■ ***Рис. 5.*** Процесс вычисления $\mu_{3:4}$

■ ***Fig. 5.*** $\mu_{3:4}$ calculating process

Полученная формула (4) является рекуррентной, и ее вычисление осуществляется с использованием метода динамического программирования. Каждой паре параметров w и n соответствует отдельная ячейка двумерного массива, в которую записывается значение $\mu_{w:n}$. Промежуточные значения $\mu_{1:n-w+1}$, $\mu_{2:n-w+2}$, …, $\mu_{w-1:n-1}$, $\mu_{w:n}$ последовательно вычисляются и сохраняются в массиве. Каждое новое значение $\mu_{w:n}$ определяется на основе ранее полученных результатов, и при его вычислении осуществляется обращение к уже заполненным ячейкам массива, что позволяет избежать повторных вычислений. В качестве иллюстрации на рис. 5 представлен процесс вычисления значения $\mu_{3:4}$:

$$\mu_{3:4} \sim \mu_{2:3} + \mu_{1:2},$$
$$\mu_{2:3} \sim \mu_{1:2}.$$

Пусть k — номер актуальной версии данных на текущий момент. Обозначим вероятность считывания последней на момент чтения версии данных запросом на чтение как $P_1 = \Pr\{\min(U_1, U_2, \ldots, U_r) = k\}$, где U_i — номер версии данных, полученных в результате i-го запроса на чтение после завершения текущего кадра. Соответственно, вероятность считывания предпоследней версии данных будет $P_2 = \Pr\{\min(U_1, U_2, \ldots, U_r) = k - 1\}$. Следовательно, в общем случае вероятность $P_i = \Pr\{\min(U_1, U_2, \ldots, U_r) = k - (i - 1)\}$, где i меняется в диапазоне от 1 до k.

Утверждение 3. Математическое ожидание среднего возраста информации $\bar{\Delta}$ для средней длины кадра $\mu_{w:n}$ при любых w и n

$$\bar{\Delta} = \left(\mu_{w:n} + \frac{c-1}{2}\right) + \sum_{i=2}^{\infty} P_i (i-1)(\mu_{w:n} + c). \quad (6)$$

Доказательство:

Пусть k — номер актуальной версии данных на текущий момент. По причине того, что система работает бесконечно долгое время, актуальная

версия может принимать любое неограниченное значение. Тогда, используя выражение (4), можно привести следующую цепочку рассуждений.

Исходя из описания работы системы, средний возраст информации в случае, когда операцией чтения считывается k-я версия данных, может быть вычислена как

$$\mu_{w:n} + \frac{c-1}{2},$$

для случая считывания $k-1$ версии данных

$$\mu_{w:n} + \frac{c-1}{2} + \mu_{w:n} + c,$$

для $k-2$ версии данных

$$\mu_{w:n} + \frac{c-1}{2} + 2\left(\mu_{w:n} + c\right).$$

Для версий $k-3$ и старее вычисление происходит аналогично.

Учитывая, что чтение каждой версии данных происходит с вероятностями P_1, P_2 и т. д., можно получить выражение для расчета среднего возраста информации

$$\overline{\Delta} = P_1\left(\mu_{w:n} + \frac{c-1}{2}\right) + P_2\left(\mu_{w:n} + \frac{c-1}{2} + \mu_{w:n} + c\right) + P_3\left(\mu_{w:n} + \frac{c-1}{2} + 2\left(\mu_{w:n} + c\right)\right) + \ldots$$

Упростим выражение:

$$\overline{\Delta} = \left(\mu_{w:n} + \frac{c-1}{2}\right) + \sum_{i=2}^{\infty} P_i\left(i-1\right)\left(\mu_{w:n} + c\right). \quad \square$$

Следует отметить, что значения вероятностей P_i вычисляются достаточно сложно. Далее будет показано, как можно построить верхнюю оценку для среднего возраста информации. С этой целью видоизменим правило работы системы: будем предполагать, что для видоизмененной системы обновление к концу кадра получает ровно w узлов. Учитывая, что в исходной системе обновление может получить больше, чем w узлов, средний возраст информации для видоизмененной системы будет больше, чем для исходной.

Обозначим через $\hat{P}_1$ вероятность считывания последней на момент чтения версии данных. В общем случае, используя рассуждения, которые вводились для исходной системы, через $\hat{P}_i$ обозначим то, что в результате r запросов на чтение номер самой актуальной версии равен $k-(i-1)$, где k — номер актуальной версии.

Утверждение 4. Для видоизмененной системы вероятности $\hat{P}_i$ рассчитываются следующим образом:

$$\hat{P}_i = q^{i-1}\left(1-q\right), \tag{7}$$

где q — вероятность того, что при чтении обращение будет относиться к узлам, на которых нет новой версии:

$$q = \prod_{i=0}^{r-1} \frac{n-w-i}{n-i}. \tag{8}$$

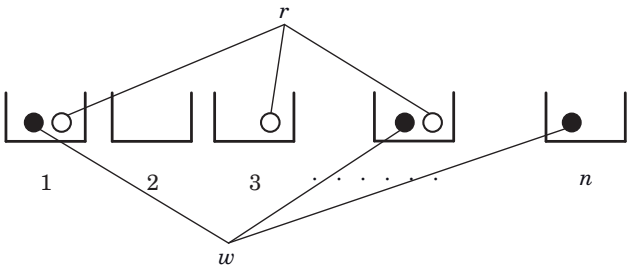
Доказательство:

Из описания работы видоизмененной модели следует, что после завершения обновления из n узлов в системе ровно w узлов будут иметь последнюю версию информации. Таким образом, если W — это множество узлов, получивших обновление, а R — это множество узлов, с которых производится чтение, то вероятность $\hat{P}_i$ можно записать как $\hat{P}_i = \Pr\{R \subset W\}$.

Рассмотрим рис. 6. Пусть имеется n ящиков. В ящики случайно помещаются w черных шаров и r белых шаров. В данной интерпретации ящики обозначают узлы, черные шары — запись нового обновления, а белые шары — операцию чтения.

Вероятность того, что первый белый шар не попадет в ящик с черным шаром, равна $\frac{n-w}{n}$; для второго белого шара $\frac{n-1-w}{n-1}$; для третьего белого шара $\frac{n-2-w}{n-2}$, и далее аналогично. Таким образом, получим полное выражение для q:

$$q = \frac{n-w}{n} \cdot \frac{n-1-w}{n-1} \cdot \frac{n-2-w}{n-2} \cdot \ldots \cdot \frac{n-(r-1)-w}{n-(r-1)} = \prod_{i=0}^{r-1} \frac{n-w-i}{n-i}. \quad \square$$



■ ***Рис. 6.*** Иллюстрация к задаче вычисления оценки снизу вероятности того, что W и R не пересекутся

■ ***Fig. 6.*** Illustration for the problem of computing a lower bound on the probability that W and R do not intersect

Используя полученную вероятность чтения последней на момент чтения версии данных (7) и выражение (6) из утверждения 3, можно получить выражение для точного вычисления среднего возраста информации для видоизмененной модели, которое также является оценкой сверху для исходной модели.

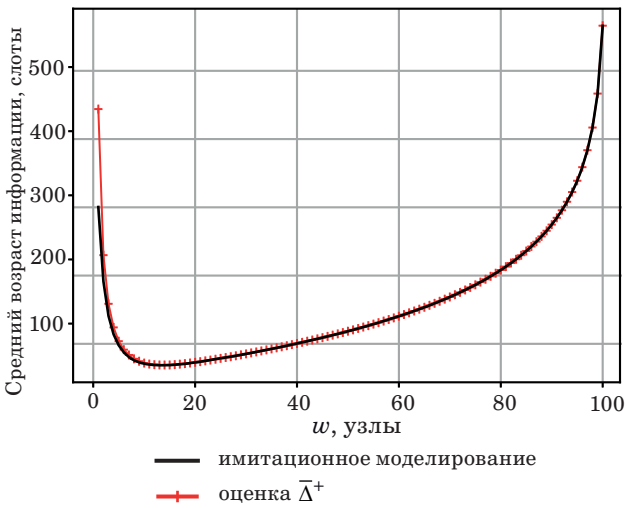
Утверждение 5. Выражение для вычисления точного значения математического ожидания среднего возраста информации $\overline{\Delta}^{+}$ для средней длины кадра $\mu_{w:n}$ при любых w и n в видоизмененной модели

$$\overline{\Delta}^{+} = \left(\mu_{w:n} + \frac{c-1}{2}\right) + \sum_{i=2}^{\infty} \hat{P}_i\left(i-1\right)\left(\mu_{w:n} + c\right). \quad (9)$$

Средний возраст информации согласно формуле (9), что является оценкой сверху для исходной модели, и средний возраст информации, полученный имитационным моделированием, показаны на рис. 7.

Из графиков видно, что значения очень близки при всех w, следующих после некоторого оптимального значения размера кворума записи, соответствующего минимуму среднего возраста информации. До оптимума же наблюдается небольшое расхождение.

Далее будет сформулирована оптимизационная задача для получения оптимального размера кворума, при котором средний возраст информации будет минимальным.



■ ***Рис. 7.*** Зависимость среднего возраста информации, полученного имитационным моделированием и его оценкой сверху, от размера кворума записи

■ ***Fig. 7.*** Graphs of the average age of information obtained by simulation modeling and its upper bound as a function of the write quorum size

Выбор размера кворума записи

При определенных значениях размера кворума записи w средний возраст информации может принимать некоторое минимальное значение. Наличие минимума объясняется тем, что при слишком малом размере кворума записи в конце кадров числа узлов, получивших последнюю версию данных, недостаточны, чтобы регулярно получать самую последнюю информацию операцией чтения, следовательно, средний возраст информации увеличивается. При слишком большом размере кворума записи увеличивается размер кадра вследствие того, что успешная запись происходит лишь с некоторой вероятностью. Этот эффект обусловлен тем, что кадр увеличивается, а значит, проходит больше времени с момента инициализации этого обновления, поэтому средний возраст также растет. Таким образом, есть некоторые оптимальные значения размера кворума записи, при которых средний возраст информации будет минимальным.

Выбор размера кворума записи можно сформулировать в виде оптимизационной задачи по минимизации среднего возраста информации, где w может принимать значения от 1 до n при заданных значениях параметров n и c:

$$w_{\text{opt}} = \arg\min_{w} \overline{\Delta}\left(w, n, c\right). \quad (10)$$

Значения среднего возраста информации $\overline{\Delta}(w, n, c)$ могут быть получены с помощью имитационного моделирования с любой заданной точностью, однако это требует больших вычислительных затрат.

Выше был описан способ вычисления верхней оценки для среднего возраста информации (9). Для верхней оценки можно сформулировать следующую оптимизационную задачу:

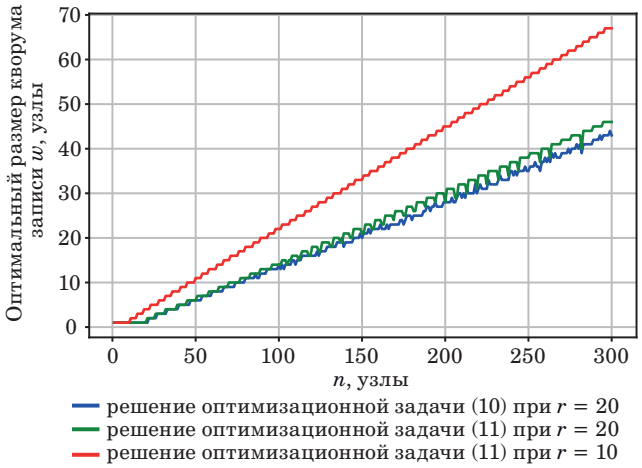
$$\hat{w}_{\text{opt}} = \arg\min_{w} \overline{\Delta}^{+}\left(w, n, c\right). \quad (11)$$

Для решения оптимизационных задач (10) и (11) можно использовать полный перебор w или одну из процедур одномерного поиска минимума, такую, как деление отрезка пополам, золотое сечение или числа Фибоначчи.

Далее будут рассмотрены результаты решения оптимизационной задачи (10) с использованием имитационного моделирования и результаты численного решения оптимизационной задачи (11).

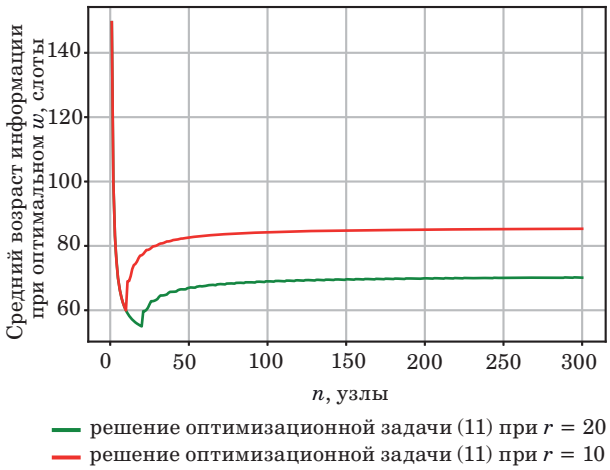
Численные результаты

Оптимальное число узлов кворума записи в зависимости от общего числа узлов в системе получено с помощью решений оптимизационных



■ ***Рис. 8.*** Зависимость оптимального числа узлов кворума записи w от общего числа узлов n в системе

■ ***Fig. 8.*** Graphs of the optimal number of write quorum nodes w, as a function of the total number of nodes in the system n



■ ***Рис. 9.*** Зависимость среднего возраста информации в условиях оптимального кворума от n

■ ***Fig. 9.*** Graphs of the average age of information under optimal quorum as a function of n

задач (10) и (11) (рис. 8). Для вычислений использовались параметры: $r = 20$, $p = 0{,}01$, $c = 100$. Отдельно были посчитаны значения для $r = 10$. Параметр n принимал значения от 1 до 300. Для получения решения оптимизационной задачи (10) использовалось моделирование с одним миллионом экспериментов.

По мере роста числа n заметен рост числа узлов кворума записи, однако зависимости не являются монотонно возрастающими. В начале графиков наблюдается оптимальное число узлов кворума записи w, равное 1, что обусловливается явлением строгого кворума. В этом случае операцией чтения всегда будет получена самая актуальная версия данных, поскольку хотя бы один узел, с которого производится чтение, имеет последнее обновление. Также наблюдается отклонение значений, полученных с помощью (10) и (11) при $r = 20$. Наибольшее расхождение между значениями не превышает трех узлов. Такое расхождение может быть вызвано близкими значениями среднего возраста информации в области его минимума. Данные расхождения практически не влияют на средний возраст информации, рассчитанный с использованием оптимальных значений размера кворума записи, полученного решениями оптимизационных задач (10) и (11). При меньшем размере кворума чтения оптимальное число узлов кворума записи растет быстрее по мере увеличения n, что можно видеть на графике при $r = 10$.

На рис. 9 рассматривается средний возраст информации в условиях оптимального кворума в зависимости от n, согласно полученным данным, показанным на рис. 8 для выражения (11).

Средний возраст информации при оптимальных значениях w, полученных имитационным моделированием для выражения (10), не показан на рис. 9, так как он практически полностью совпадает в данном масштабе графиков с выражением (11). Максимальное расхождение между данными значениями среднего возраста информации не превышает 0,7, что можно считать несущественным. У графиков наблюдается выраженный минимум при $n = 20$ для графика с $r = 20$. До данного значения $r = n = 20$ при малом n дольше набираются необходимые w успешных подтверждений записи, из-за чего растягивается средняя длина кадра, а это влечет больший средний возраст информации. Когда r становится меньше n, то появляется вероятность, что среди случайно выбранных r узлов не окажется ни одного узла с последней версией данных, из-за чего также может повышаться средний возраст информации. Следовательно, образуется минимум, который видно на графике. Для графика при $r = 10$ аналогично.

Таким образом, наглядно продемонстрированы основные особенности работы системы в условиях оптимального кворума записи в зависимости от других параметров системы.

Заключение

Рассмотрена реплицируемая система хранения данных на основе равноправных узлов. Отмечено, что отличительной особенностью такой системы является наличие так называемых кворумов чтения и записи, которые представляют множества узлов, с которых происходит считывание данных

и запись обновления соответственно. Предложена модель, отражающая основные особенности реплицируемой системы хранения данных с кворумами чтения и записи. В предложенной модели время работы разделено на кадры и слоты, а запись обновления на узлы происходит с вероятностью p и завершается при получении не менее w подтверждений. В качестве показателя актуальности данных рассмотрен средний возраст информации, получаемый запросом на чтение. Показано, что длина кадра является случайной величиной, распределение которой соответствует w-й порядковой статистике из n независимых геометрически распределенных случайных величин с параметром p. Для ее математического ожидания выведена рекуррентная формула и предложен метод динамического программирования. Сформулирована оптимизационная задача выбора размера кворума записи, минимизирующего средний возраст информации. На основе исходной модели предложена видоизмененная модель, в которой к концу кадра обновление записывается ровно на w узлов, что позволило упростить вычисления и получить оценку сверху для среднего возраста информации. Численные результаты показали, что оптимальный размер кворума записи растет с увеличением общего числа узлов, а оценка сверху хорошо согласуется с имитационным моделированием; выявлено наличие минимума среднего возраста информации при оптимальном кворуме записи в зависимости от общего числа узлов в системе. Модель может быть использована при проектировании реплицируемых систем хранения данных для соблюдения баланса согласованности и доступности. Важным допущением является независимость успешной доставки обновления на разные узлы; в реальных системах при многоадресной передаче это может не выполняться, что требует дальнейшего исследования и учета зависимости между событиями доставки.

Финансовая поддержка

Исследование выполнено при финансовой поддержке Российского научного фонда, грант № 22-19-00305 «Пространственно-временные стохастические модели беспроводных сетей с большим числом абонентов».

Литература

1. **Богатырев В. А., Богатырев С. В., Богатырев А. В.** Граничная оценка надежности кластерных систем на основе декомпозиции марковской модели при ограниченном восстановлении узлов с накоплением отказов. *Научно-технический вестник информационных технологий, механики и оптики*, 2025, т. 25, № 3, с. 574–583. doi:10.17586/2226-1494-2025-25-3-574-583, EDN: QQJPSJ
2. **Bogatyrev V. A., Bogatyrev S. V., Bogatyrev A. V.** Recovery of real-time clusters with the division of computing resources into the execution of functional queries and the restoration of data generated since the last backup. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2024, vol. 14123, pp. 236–250. doi:10.1007/978-3-031-50482-2_19
3. **Najjar A., Mokadem R., Pierson J. M.** A review of data placement and replication strategies based on machine learning. *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, 2024, pp. 278–285. doi:10.1109/ICPADS63004.2024.00042
4. **Mastromauro L., Andrade D. S., Ozmen M. O., Kinsy M. A.** Survey of attacks and defenses on consensus algorithms for secure data replication in distributed systems. *IEEE Access*, 2025, vol. 13, pp. 143631–143667. doi:10.1109/ACCESS.2025.3597260
5. **Sevin A.** Implementation of a data-parallel approach on a lightweight hash function for IoT devices. *Mathematics*, 2025, vol. 13, no. 5, pp. 734. doi:10.3390/math13050734
6. **Noor S., Awan H. H., Hashmi A. S., Saeed A., Khan S., AlQahtani S. A.** Optimizing performance of parallel computing platforms for large-scale genome data analysis. *Computing*, 2025, vol. 107, no. 3, pp. 86. doi:10.1007/s00607-025-01441-y
7. **Patel P.** Dynamic CAP optimization in distributed databases via adaptive graph neural networks with causal inference. *Engineering Archive*, 2025, pp. 1–14. doi:10.31224/4825
8. **Borrill P.** What distributed computing got wrong: The category mistake that turned design choices into laws of nature. *arXiv preprint*, 2026, arXiv:2602.18723, pp. 1–28. doi:10.48550/arXiv.2602.18723
9. **Wright C. S.** Resolving CAP through automata-theoretic economic design: A unified mathematical framework for real-time partition-tolerant systems. *arXiv preprint*, 2025, arXiv:2507.02464, pp. 1–51. doi:10.48550/arXiv.2507.02464
10. **Song Y., Liu C., Shou M. Z.** Omniconsistency: Learning style-agnostic consistency from paired stylization data. *Advances in Neural Information Processing Systems*, 2026, vol. 38, pp. 173300–173322. doi:10.52202/085713-5216
11. **Olivieri L., Negrini L., Arceri V., Ferrara P., Cortesi A.** Detection of read-write issues in hyperledger fabric smart contracts. *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, 2025, pp. 329–337. doi:10.1145/3672608.3707721

12. **Desai M., Quinn R., Asadinia M.** SMART-WRITE: adaptive learning-based write energy optimization for phase change memory. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2025, pp. 00640–00648. doi:10.1109/CCWC62904.2025.10903957
13. **Behrouzi-Far A., Soljanin E., Yates R. D.** Data freshness in leader-based replicated storage. *2020 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2020, pp. 1806–1811. doi:10.1109/ISIT44484.2020.9174411
14. **Крылов Д. Р., Пойманова Е. Д., Тюрликов А. М.** Модель реплицируемой системы хранения данных с использованием среднего возраста информации в качестве показателя актуальности данных. *Информационно-управляющие системы*, 2024, № 3, с. 11–23. doi:10.31799/1684-8853-2024-3-11-23, EDN: XSSHJI
15. **Elhemali M., Gallagher N., Gordon N., Idziorek J., Krog R., Lazier C., Mo E., Mritunjai A., Perianayagam S., Rath T., Sivasubramanian S., Sorenson J. C. III, Sosothikul S., Terry D., Vig A.** Amazon DynamoDB: A scalable, predictably performant, and fully managed NoSQL database service. *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, 2022, pp. 1037–1048.
16. **Sagan M., Plechawska-Wójcik M.** Comparative analysis of non-relational databases on the example of Amazon DynamoDB and MongoDB. *Journal of Computer Sciences Institute*, 2026, vol. 39, pp. 108–114. doi:10.35784/jcsi.8549
17. **Golec M., Zhang Z., Wang H., Golec M., Singh R., Gill S. S.** Google cloud spanner-based data management application for smart gaming industry. *Computational Intelligence Techniques for 5G Enabled IoT Networks*, Cham, Springer Nature Switzerland, 2025, pp. 49–59. doi:10.1007/978-3-031-82733-4_4
18. **Zhu X., Nie X., Liu J.** Time series database optimization based on InfluxDB. *2023 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC)*, IEEE, 2023, pp. 879–885. doi:10.1109/PEEEC60561.2023.00172
19. **Ruan C., Zhang Y., Zhang J., Li C., Ma X., Chen H., Zhou J., Li F., Yang X.** PolyBase: Adapting to data affinity changes in geo-replicated database via row-level consensus-group affiliation re-assignment. *Proceedings of the VLDB Endowment*, 2024, vol. 18, no. 3, pp. 702–714. doi:10.14778/3712221.3712236
20. **Naseri Seyedi Noudoust N., Adabi S., Rezaee A.** A quorum-based data consistency approach for non-relational database. *Cluster Computing*, 2022, vol. 25, no. 2, pp. 1515–1540. doi:10.1007/s10586-021-03531-w
21. **Zhong J., Yates R. D., Soljanin E.** Minimizing content staleness in dynamo-style replicated storage systems. *IEEE INFOCOM 2018 – IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2018, pp. 361–366. doi:10.1109/INFCOMW.2018.8406971
22. **Yates R. D., Sun Y., Brown D. R., Kaul S. K., Modiano E., Ulukus S.** Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 2021, vol. 39, no. 5, pp. 1183–1210. doi:10.1109/JSAC.2021.3065072
23. **Qu Y., Chen Z., Pappas N., Tang C., Wang M., Quek T. Q. S.** Analysis of age of information for a discrete-time dual-queue system. *IEEE Transactions on Communications*, 2025, vol. 73, no. 11, pp. 10855–10870. doi:10.1109/TCOM.2025.11015332
24. **Zhang C., Wang L., Xiao L., Jiang S., Han M., Wang J., Wei B., Qin G.** Minimizing the cost of periodically replicated systems via model and quantitative analysis. *Frontiers of Computer Science*, 2024, vol. 18, no. 5, art. 185206. doi:10.1007/s11704-023-2625-8

UDC 004.62, 004.7
doi:10.31799/1684-8853-2026-5-73-84
EDN: CWRBDL

Average age of information in distributed data storage systems with peer nodes

D. R. Krylov[a], Post-Graduate Student, orcid.org/ 0009-0008-5901-5342
U. A. Bolotskikh[a], Master Student, orcid.org/0009-0009-0733-594X
A. M. Turlikov[a], Dr. Sc., Tech., Professor, orcid.org/0000-0001-7132-094X, turlikov@vu.spb.ru
[a]Saint-Petersburg State University of Aerospace Instrumentation, 67, B. Morskaia St., 190000, Saint-Petersburg, Russian Federation

Introduction: Distributed data storage systems can employ leader and follower nodes or peer nodes. For systems with peer nodes, researchers have given little attention to how system parameters affect a performance metric such as average age of information; therefore, investigating this effect is a pressing task. **Purpose:** To develop a model of a replicated data storage system with peer nodes that allows formulating and solving the problem of minimizing average age of information by choosing write quorum size. **Results:** We propose a model that reflects the main operational features of replicated data storage systems with peer nodes, to which the system multicasts an update over an unreliable channel. Successful update delivery events occur independently for different nodes. One of the model parameters is the probability of successful update delivery. We specify a method for calculating an upper bound for the average age of information. We compare this bound with simulation modeling and show that it differs from the exact value by no more than 1%. We formulate and solve an optimization problem that selects the write quorum size to minimize the average age of information. Numerical results show that the optimal write quorum size grows as the total number of nodes increases. We obtain the dependence of the average age of information on the total number of nodes for the optimal write quorum size and fixed other model parameters. We note that this dependence has a minimum. **Practical relevance:** The proposed model reflects the general features of replicated data storage systems with peer nodes, and engineers can apply it when designing real systems to maintain a balance between consistency and availability. **Discussion:** The model relies on a key assumption: successful update delivery to different nodes is independent. In real systems with multicast transmission this assumption may fail, so further research should account for dependence between delivery events.

Keywords — average age of information, data storage, replication, read quorum, write quorum.

For citation: Krylov D. R., Bolotskikh U. A., Turlikov A. M. Average age of information in distributed data storage systems with peer nodes. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2026, no. 5, pp. 73–84 (In Russian). doi:10.31799/1684-8853-2026-5-73-84, EDN: CWRBDL

Financial support

The research was carried out with financial support from the Russian Science Foundation, grant No. 22-19-00305 “Spatiotemporal stochastic models of wireless networks with a large number of subscribers”.

References

1. Bogatyrev V. A., Bogatyrev S. V., Bogatyrev A. V. Boundary estimation of the reliability of cluster systems based on the decomposition of the Markov model with limited recovery of nodes with accumulated failures. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2025, vol. 25, no. 3, pp. 574–583. doi:10.17586/2226-1494-2025-25-3-574-583, EDN: QQJPSJ
2. Bogatyrev V. A., Bogatyrev S. V., Bogatyrev A. V. Recovery of real-time clusters with the division of computing resources into the execution of functional queries and the restoration of data generated since the last backup. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2024, vol. 14123, pp. 236–250. doi:10.1007/978-3-031-50482-2_19
3. Najjar A., Mokadem R., Pierson J. M. A review of data placement and replication strategies based on machine learning. *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, IEEE, 2024, pp. 278–285. doi:10.1109/ICPADS63004.2024.00042
4. Mastromauro L., Andrade D. S., Ozmen M. O., Kinsy M. A. Survey of attacks and defenses on consensus algorithms for secure data replication in distributed systems. *IEEE Access*, 2025, vol. 13, pp. 143631–143667. doi:10.1109/ACCESS.2025.3597260
5. Sevin A. Implementation of a data-parallel approach on a lightweight hash function for IoT devices. *Mathematics*, 2025, vol. 13, no. 5, pp. 734. doi:10.3390/math13050734
6. Noor S., Awan H. H., Hashmi A. S., Saeed A., Khan S., AlQahtani S. A. Optimizing performance of parallel computing platforms for large-scale genome data analysis. *Computing*, 2025, vol. 107, no. 3, pp. 86. doi:10.1007/s00607-025-01441-y
7. Patel P. Dynamic CAP optimization in distributed databases via adaptive graph neural networks with causal inference. *Engineering Archive*, 2025, pp. 1–14. doi:10.31224/4825
8. Borrill P. What distributed computing got wrong: The category mistake that turned design choices into laws of nature. *arXiv preprint*, 2026, arXiv:2602.18723, pp. 1–28. doi:10.48550/arXiv.2602.18723
9. Wright C. S. Resolving CAP through automata-theoretic economic design: A unified mathematical framework for real-time partition-tolerant systems. *arXiv preprint*, 2025, arXiv:2507.02464, pp. 1–51. doi:10.48550/arXiv.2507.02464
10. Song Y., Liu C., Shou M. Z. Omniconsistency: Learning style-agnostic consistency from paired stylization data. *Advances in Neural Information Processing Systems*, 2026, vol. 38, pp. 173300–173322. doi:10.52202/085713-5216
11. Olivieri L., Negrini L., Arceri V., Ferrara P., Cortesi A. Detection of read-write issues in hyperledger fabric smart contracts. *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, 2025, pp. 329–337. doi:10.1145/3672608.3707721
12. Desai M., Quinn R., Asadinia M. SMART-WRITE: adaptive learning-based write energy optimization for phase change memory. *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2025, pp. 00640–00648. doi:10.1109/CCWC62904.2025.10903957
13. Behrouzi-Far A., Soljanin E., Yates R. D. Data freshness in leader-based replicated storage. *2020 IEEE International Symposium on Information Theory (ISIT)*, IEEE, 2020, pp. 1806–1811. doi:10.1109/ISIT44484.2020.9174411
14. Krylov D. R., Poymanova E. D., Turlikov A. M. Modeling a replicated storage system with the use of the average age of information as an indicator of data relevance. *Informatsionno-upravliaiushchie sistemy* [Information and Control Systems], 2024, no. 3, pp. 11–23 (In Russian). doi:10.31799/1684-8853-2024-3-11-23, EDN: XSSHJI
15. Elhemali M., Gallagher N., Gordon N., Idziorek J., Krog R., Lazier C., Mo E., Mritunjai A., Perianayagam S., Rath T., Sivasubramanian S., Sorenson J. C. III, Sosothikul S., Terry D., Vig A. Amazon DynamoDB: A scalable, predictably performant, and fully managed NoSQL database service. *2022 USENIX Annual Technical Conference (USENIX ATC 22)*, 2022, pp. 1037–1048.
16. Sagan M., Plechawska-Wójcik M. Comparative analysis of non-relational databases on the example of Amazon DynamoDB and MongoDB. *Journal of Computer Sciences Institute*, 2026, vol. 39, pp. 108–114. doi:10.35784/jcsi.8549
17. Golec M., Zhang Z., Wang H., Golec M., Singh R., Gill S. S. Google cloud spanner-based data management application for smart gaming industry. *Computational Intelligence Techniques for 5G Enabled IoT Networks*, Cham, Springer Nature Switzerland, 2025, pp. 49–59. doi:10.1007/978-3-031-82733-4_4
18. Zhu X., Nie X., Liu J. Time series database optimization based on InfluxDB. *2023 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC)*, IEEE, 2023, pp. 879–885. doi:10.1109/PEEEC60561.2023.00172
19. Ruan C., Zhang Y., Zhang J., Li C., Ma X., Chen H., Zhou J., Li F., Yang X. PolyBase: Adapting to data affinity changes in geo-replicated database via row-level consensus-group affiliation re-assignment. *Proceedings of the VLDB Endowment*, 2024, vol. 18, no. 3, pp. 702–714. doi:10.14778/3712221.3712236
20. Naseri Seyedi Noudoust N., Adabi S., Rezaee A. A quorum-based data consistency approach for non-relational database. *Cluster Computing*, 2022, vol. 25, no. 2, pp. 1515–1540. doi:10.1007/s10586-021-03531-w
21. Zhong J., Yates R. D., Soljanin E. Minimizing content staleness in dynamo-style replicated storage systems. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, IEEE, 2018, pp. 361–366. doi:10.1109/INFCOMW.2018.8406971
22. Yates R. D., Sun Y., Brown D. R., Kaul S. K., Modiano E., Ulukus S. Age of information: An introduction and survey. *IEEE Journal on Selected Areas in Communications*, 2021, vol. 39, no. 5, pp. 1183–1210. doi:10.1109/JSAC.2021.3065072
23. Qu Y., Chen Z., Pappas N., Tang C., Wang M., Quek T. Q. S. Analysis of age of information for a discrete-time dual-queue system. *IEEE Transactions on Communications*, 2025, vol. 73, no. 11, pp. 10855–10870. doi:10.1109/TCOM.2025. 11015332
24. Zhang C., Wang L., Xiao L., Jiang S., Han M., Wang J., Wei B., Qin G. Minimizing the cost of periodically replicated systems via model and quantitative analysis. *Frontiers of Computer Science*, 2024, vol. 18, no. 5, art. 185206. doi:10.1007/s11704-023-2625-8