

УДК 004.75:004.424.47
DOI 10.17513/snt.40813



МЕТОД АДАПТИВНОГО КОНСИСТЕНТНОГО ХЕШИРОВАНИЯ С БЮДЖЕТНЫМ ОГРАНИЧЕНИЕМ СТОИМОСТИ ПЕРЕСТРОЙКИ РАСПРЕДЕЛЕНИЯ КЛЮЧЕЙ

Батанов А. О. ORCID ID 0009-0004-6968-1139, Болбаков Р. Г.

*Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет», Москва, Российская Федерация,
e-mail: batanov@mirea.ru*

Рассматривается задача балансировки нагрузки в распределенных системах хранения данных с гетерогенными узлами. Применяемые подходы на основе консистентного хеширования либо не учитывают изменение нагрузки во времени, либо не ограничивают стоимость перестройки распределения, что ведет к избыточным миграциям данных. Цель исследования – разработать метод адаптивного управления распределением ключей, обеспечивающий снижение дисбаланса нагрузки при соблюдении ограничений на стоимость перестройки. Предложен метод, включающий коэффициент интенсивности управления, пошаговый бюджет переназначений токенов и гистерезис порогов чувствительности для подавления ложных срабатываний при шумной телеметрии. Разработана имитационная модель кластера с параметризованной гетерогенностью узлов трех классов, нестационарной пуассоновской нагрузкой, распределением ключей Ципфа и зашумленной телеметрией. Проведен вычислительный эксперимент из 7 серий по 30 повторений каждая с проверкой инвариантов корректности на каждом шаге. По результатам вычислительного эксперимента зафиксировано статистически значимое снижение дисбаланса нагрузки относительно алгоритма равномерно-го распределения в стационарных и нестационарных сценариях. В стационарных условиях качество балансировки сопоставимо с алгоритмом статического взвешивания, требующим знания емкостей узлов заранее. При динамическом изменении состава кластера снижение дисбаланса статистически значимо относительно обоих статических алгоритмов. Стоимость перестройки ниже, чем у алгоритма динамического взвешивания без бюджетного ограничения. Установлены условия, при которых применение метода нецелесообразно: совокупное влияние значительного шума, задержки и пропусков телеметрии приводит к избыточной миграционной активности без соответствующего снижения дисбаланса.

Ключевые слова: консистентное хеширование, распределенные системы, балансировка нагрузки, адаптивное управление, гетерогенные кластеры, миграция данных, бюджет перестройки

ADAPTIVE CONSISTENT HASHING METHOD WITH BUDGETED REDISTRIBUTION COST CONSTRAINT

Batanov A. O. ORCID ID 0009-0004-6968-1139, Bolbakov R. G.

*Federal State Budgetary Educational Institution of Higher Education
“MIREA – Russian Technological University”, Moscow, Russian Federation,
e-mail: batanov@mirea.ru*

This paper considers the problem of load balancing in distributed data storage systems with heterogeneous nodes. Existing consistent hashing approaches either do not account for load dynamics over time or do not constrain redistribution costs, leading to excessive data migrations. The aim of this study is to develop an adaptive key distribution method that reduces load imbalance subject to constraints on redistribution cost. The proposed method employs a control intensity coefficient, a per-step token reassignment budget, and hysteresis-based sensitivity thresholds to suppress spurious activations under noisy telemetry. A simulation model was developed incorporating heterogeneous nodes, non-stationary workload, and telemetry noise. A computational experiment across seven series showed a statistically significant reduction in load imbalance in both stationary and non-stationary scenarios. Under stationary conditions, balancing quality is comparable to a static weighted algorithm that requires prior knowledge of node capacities. Under dynamic cluster membership changes, the reduction in imbalance is statistically significant relative to both static algorithms. Redistribution cost is lower than that of the dynamic reweighting algorithm without budget constraints. Conditions under which application of the method is inadvisable were identified: concurrent high noise, telemetry lag, and measurement gaps lead to excessive migration activity without a corresponding reduction in imbalance.

Keywords: consistent hashing, distributed systems, load balancing, adaptive control, heterogeneous clusters, data migration, redistribution budget

Введение

В распределенных системах хранения данных назначение ключей узлам осуществляется через консистентное хеширование: каждому ключу сопоставляется позиция на хеш-кольце, а ближайший токен опреде-

ляет ответственный узел [1]. Данный механизм используется в промышленных системах распределенного хранения данных [2, 3]. Он также применяется в одноранговых протоколах [4] и сетевых балансировщиках нагрузки [5]. Масштабируемые варианты

консистентного хеширования направлены на снижение стоимости переназначений при изменении состава множества узлов [6].

При неоднородности узлов по ресурсным емкостям равномерное распределение токенов приводит к дисбалансу нагрузки: слабые узлы перегружены, сильные простаивают. Статическое взвешивание через виртуальные узлы, применяемое в схемах горизонтального масштабирования [7], не адаптируется к динамике нагрузки. Динамическое взвешивание по наблюдаемой нагрузке, рассматриваемое в работах по балансировке неоднородных систем [8, 9], может быть чувствительно к колебаниям параметров распределения и само по себе не задает бюджета перестройки. Современные работы по балансировке нагрузки в гетерогенных вычислительных средах также фиксируют компромисс между учетом текущей нагрузки, затратами на взаимодействие узлов и устойчивостью планирования [10]. Метод *bounded loads* [11] ограничивает нагрузку перенаправлением запросов, но не управляет числом токенов и не формализует ограничения на объем миграций.

Каждое переназначение токенов влечет миграцию данных между узлами, что создает дополнительную нагрузку на сетевую подсистему и подсистему хранения [2, 12]. Исследования распределенных систем и оптимизации ресурсов баз данных показывают, что маршрутизация запросов, репликация, миграции данных и балансировка соединений создают самостоятельные накладные расходы [12, 13]. Отсутствие формализованных ограничений на объем миграций приводит к режиму непрерывной перестройки, ухудшающему качество обслуживания [9, 14]. В работах по балансировке при гетерогенных узлах [7, 14] используются виртуальные токены и квантовые весовые схемы, однако бюджет перестройки не формализуется. Подход на основе динамических весов с эвристической оптимизацией [9] снижает дисбаланс, но не ограничивает накопленную стоимость миграций. Ни в одном из перечисленных подходов ограничения на стоимость адаптации не формализованы явно.

Среди современных алгоритмов консистентного хеширования с малым потреблением памяти выделяются *MementoHash* [15], *DxHash* [16] и *JumpHash* [17]. По данным сравнительного обзора [18], *remigrate* этих алгоритмов близок к теоретическому минимуму K/N (K – число ключей, N – число узлов): при добавлении или удалении узла мигрирует приблизительно $1/N$ доли ключевого пространства. По памяти *MementoHash* и *DxHash* требуют $O(N)$,

JumpHash – $O(1)$; по сложности поиска *JumpHash* и *DxHash* – $O(1)$, *MementoHash* – $O(\log N)$ в среднем. В базовой постановке и в проверенных источниках эти алгоритмы не формализуют адаптацию к разным текущим емкостям узлов; поэтому они не закрывают задачу распределения нагрузки между узлами различной емкости. Смежные направления включают динамическую балансировку на основе хеш-слотов [19], консистентное хеширование с динамическими весами [9] и модификацию равномерности кольца консистентного хеширования [20], а также аналитическое описание сценариев LRU-кэширования при консистентном хешировании [21]. Альтернативный подход в *Pegasus* [22] основан на избирательной репликации горячих ключей через программируемые сетевые коммутаторы. Метод АСН дополняет перечисленные работы: он управляет назначением токенов гетерогенных узлов на хеш-кольце при ограничении стоимости перестройки и совместим с указанными схемами минимального переназначения.

Цель исследования – разработать метод адаптивного управления распределением ключей на основе консистентного хеширования, при котором дисбаланс нагрузки снижается в рамках заданных ограничений на стоимость перестройки.

Материалы и методы исследования

Предлагаемый метод адаптивного консистентного хеширования (АСН) функционирует на уровне маршрутизации запросов между клиентом и узлами кластера. Схема встраивания в архитектуру распределенной системы показана на рис. 1: метод размещается между клиентом и узлами кластера, получает телеметрию от узлов и управляет назначением токенов хеш-кольца. Метод включает три конструктивных элемента.

Элемент 1. Коэффициент интенсивности управления. Коэффициент интенсивности $C(t) = C_{\max} \cdot \gamma(t) \cdot \eta(t)$ задает масштаб вмешательства на уровне кластера. Коэффициент разрешенности перестройки $\gamma(t) = \text{clip}((\tau - \text{total_load}(t)) / (\tau - \tau_{\min}), 0, 1)$ подавляет перестройку при приближении агрегированной загрузки кластера к насыщению. Показатель локального дисбаланса $\eta(t) = I(t) / (I(t) + \kappa)$ связывает интенсивность управления с масштабом перегрузки, где $I(t)$ – максимальное положительное отклонение загрузки узла от целевого уровня ℓ^* . Мультипликативная форма $C(t)$ приводит к нулевой интенсивности управления в двух случаях: при отсутствии локальной перегрузки ($\eta = 0$) и при насыщении кластера ($\gamma = 0$) [23].

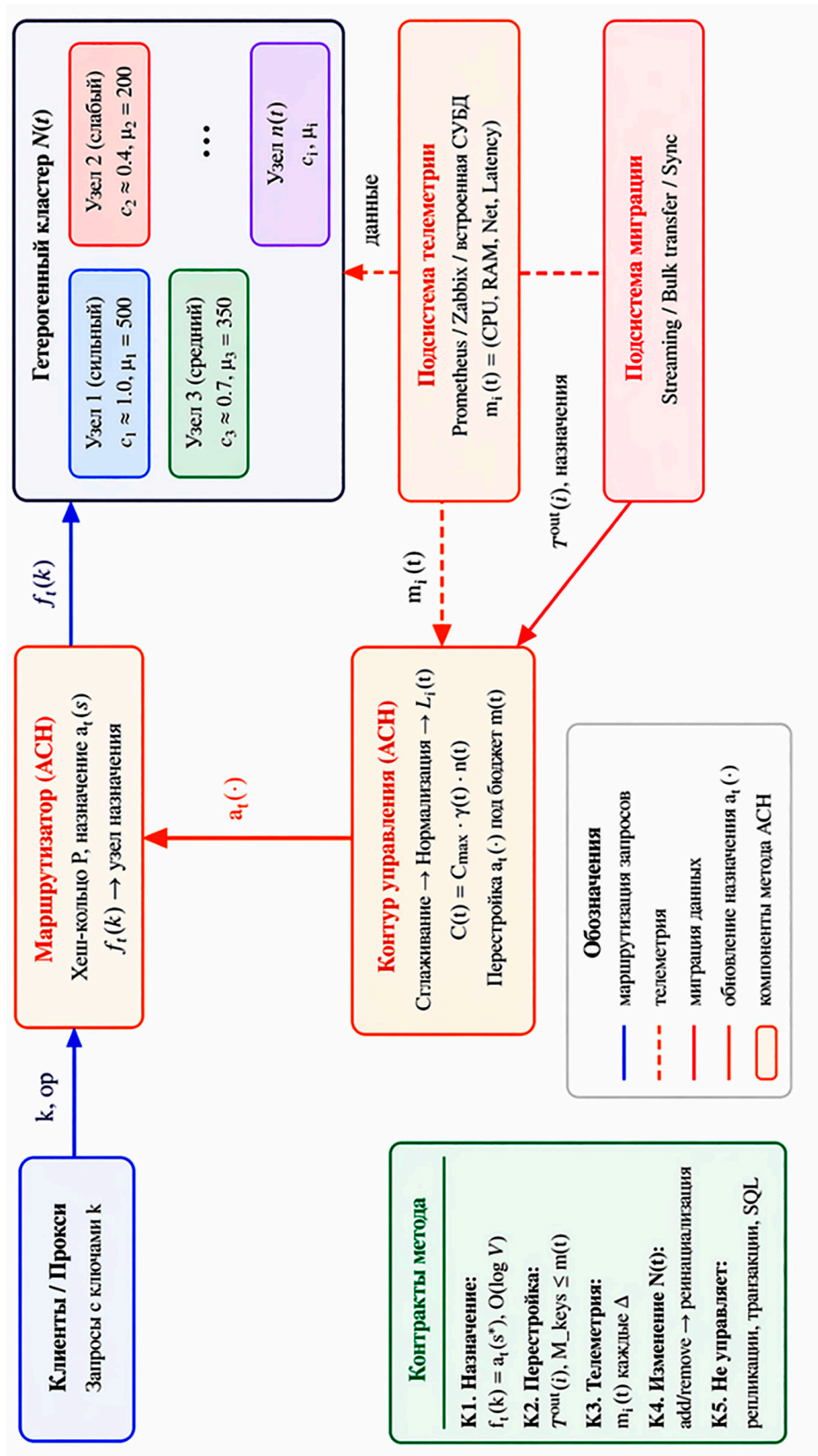


Рис. 1. Архитектурная схема встраивания метода АСН в распределенную систему управления данными.
Примечание: составлен авторами по результатам исследования

Элемент 2. Бюджет переназначений. Позиции токенов на хеш-кольце фиксированы; адаптация выполняется через переназначение токенов между узлами. Стоимость перестройки определяется как доля переназначенного ключевого пространства: $M_keys(t) = \sum L_s$ по токенам с измененным назначением, где L_s – длина сегмента хеш-пространства. Бюджет переназначений $m(t) = \min(M_max, \Delta/N \cdot M_H) \cdot C(t)/C_max$ ограничивает плановый объем переназначений; из-за дискретной длины токенов фактическая величина дополнительно контролируется инвариантом

$$M_keys(t) \leq m(t) + |O| \cdot \max_s L_s.$$

Двойное ограничение (пошаговое M_max и горизонтное M_H) ограничивает как одиночные крупные перестройки, так и накопление миграционной активности на горизонте [23].

Элемент 3.

Гистерезис порогов чувствительности. Дискретное состояние $\sigma_i(t) \in \{-1, 0, +1\}$ определяет, включен ли узел в множество ис-

точников (перегруженных, $\sigma = +1$) или приемников (недогруженных, $\sigma = -1$). Пороги $\varepsilon_on > \varepsilon_off > 0$ обеспечивают запас устойчивости: переключение σ_i из нейтрального состояния ($\sigma = 0$) требует отклонения загрузки не менее ε_on , а возврат в нейтральное состояние – снижения отклонения до ε_off . При шуме телеметрии $|\xi_i(t)| \leq \varepsilon_off$ состояние σ_i остается нейтральным и перестройка не инициируется [23].

Для экспериментального исследования разработана имитационная модель кластера из 10 узлов трех классов гетерогенности (табл. 1). Модель включает пуассоновский поток запросов с интенсивностью $\Lambda(t)$, распределение ключей маршрутизации по закону Ципфа с параметром s , модель насыщения ресурсов (очередь M/M/1) и модель искажений телеметрии (аддитивный шум σ_v , лаг d шагов, пропуски с вероятностью p_miss). Интенсивность $\Lambda = 1437,5$ запросов в шаг определена калибровкой, при которой слабые узлы перегружены ($\ell_i > \ell^* + \varepsilon_on = 0,75$), а сильные недогружены ($\ell_i < \ell^* - \varepsilon_on = 0,55$).

Таблица 1

Параметры классов узлов в имитационной модели

Параметр	Класс А (сильный)	Класс В (средний)	Класс С (слабый)
c_cpu (отн. ед.)	1,0	0,55	0,25
μ_i (запр./шаг)	350	193	88
Доля в кластере	30 %	40 %	30 %

Примечание: составлена авторами на основе полученных данных в ходе исследования.

Алгоритм 1. Перестройка распределения токенов на шаге t

Вход: кольцо R , оценки загрузки ℓ_i , шаг t
состояния $\sigma_i \in \{-1, 0, +1\}$

Выход: обновленное кольцо R , $M_keys(t)$

1. обновить σ_i по правилам гистерезиса ($\varepsilon_on, \varepsilon_off$)
2. $C_norm \leftarrow \gamma(t) \cdot \eta(t)$ // $C_norm = C(t) / C_max$
3. $m \leftarrow \min(M_max, M_H / H) \cdot C_norm$
4. если $C \leq 0$ или $m \leq 0$ – возврат 0
5. $O \leftarrow \{i : \sigma_i = +1\}$; $U \leftarrow \{j : \sigma_j = -1\}$
6. $g_i \leftarrow \max(\ell_i - \ell^*, 0)$ для $i \in O$
7. $r_j \leftarrow \max(\ell^* - \ell_j, 0)$ для $j \in U$
8. если $\sum g_i = 0$ или $\sum r_j = 0$ – возврат 0
9. $T_out \leftarrow \emptyset$ // Фаза 1
10. для каждого $i \in O$:
 11. $share_i \leftarrow m \cdot g_i / \sum_k g_k$
 12. отсортировать токены узла i по убыванию L_s
 13. набирать токены в T_out , пока $\sum L_s < share_i$
 14. остановка по достижении v_min оставшихся токенов
15. для каждого $j \in U$: // Фазы 2 – 3
 16. $n_j \leftarrow \lfloor |T_out| \cdot r_j / \sum_k r_k \rfloor$
 17. переназначить n_j токенов из T_out на узел j
18. $M_keys(t) \leftarrow \sum L_s$ по переназначенным токенам
19. проверить инвариант: $M_keys(t) \leq m + |O| \cdot \max_s L_s$
20. возврат $M_keys(t)$

Таблица 1а

Параметры метода АСН, используемые в экспериментах

Параметр	Значение	Назначение
ℓ^*	0,65	Целевой уровень загрузки узла
ε_{on}	0,10	Порог включения коррекции для узла
ε_{off}	0,04	Порог выключения коррекции (запас гистерезиса)
τ	0,85	Верхний порог подавления перестройки при насыщении кластера
τ_{min}	0,50	Нижний порог насыщения; ниже него $\gamma(t) = 1$
κ	0,10	Смещение в показателе локального дисбаланса $\eta(t)$
M_{max}	0,02	Максимальная пошаговая доля переназначенного пространства
M_H	0,25	Бюджет переназначений на горизонте H шагов
H	100	Длина горизонтного окна (шагов)
v_{min}	3	Минимальное число токенов на узле

Примечание: составлена авторами на основе полученных данных в ходе исследования; значения соответствуют программной реализации имитационной модели.

Таблица 2

Сводные результаты метода АСН по сериям С1–С5 (30 повторений)

Серия	Сценарий	$D(ACH)$	$D(A0)$	Снижение	M_{cum}	n_{viol}
C1	Стационарный	0,177	0,225	-21,5 %	0,063	0
C2	Нестационарный	0,016	0,035	-55,4 %	0,028	0
C3	Ципф + churn	0,340	0,496	-31,6 %	0,310	0
C4a	Шум $\sigma = 0,02$	0,172	0,224	-23,2 %	0,070	0
C4b	Шум $\sigma = 0,05$	0,166	0,222	-25,2 %	0,167	0
C4c	Шум + лаг	0,198	0,219	-9,8 %	4,565	0

Примечание: D – средний дисбаланс на окне анализа [40, 2000), M_{cum} – накопленная стоимость перестройки, n_{viol} – число нарушений инвариантов корректности. Критерий Вилкоксона: $p < 0,001$ для серий С1–С4с.

Составлена авторами на основе полученных данных в ходе исследования.

Процедура переназначения токенов реализована в три фазы. На первой фазе для каждого узла-источника ($\sigma_i = +1$) токены упорядочиваются по убыванию длины дуги L_s и отбираются до достижения индивидуальной доли бюджета

$$share_i = m(t) \cdot g_i / \sum_j g_j,$$

где $g_i = \max(\ell_i - \ell^*, 0)$; отбор прекращается также по достижении минимума v_{min} токенов на узле. На второй фазе отобранные токены объединяются в общий пул. На третьей фазе пул распределяется между узлами-приемниками ($\sigma_i = -1$) пропорционально дефициту $r_i = \max(\ell^* - \ell_i, 0)$. Аддитивный гауссовский шум применяется к телеметрии с одинаковой реализацией для всех сравниваемых алгоритмов на каждом шаге, что обеспечивает корректность парного критерия Вилкоксона; лаг моделируется задержкой наблюдения на d шагов, пропуски – независимыми событиями с вероятностью

p_{miss} и замещением последним валидным значением. Для каждой серии выполняется 30 повторений с воспроизводимыми начальными условиями генераторов псевдослучайных чисел; на каждом шаге каждого повторения контролируется выполнение пяти инвариантов корректности.

Полная программная реализация имитационной модели, конфигурации серий, исходные значения seed-ов, агрегированные выборки результатов, индивидуальные значения по 30 повторениям для построения boxplot и ECDF, расчетные скрипты для критерия Вилкоксона и дельты Клиффа, а также набор из 57 модульных тестов опубликованы в открытом репозитории: <https://github.com/Kronv3rk/ach-experiment> (MIT License).

Эксперимент включает семь серий: стационарный режим (С1), нестационарная нагрузка с деградацией узлов (С2), изменение состава узлов с распределением Ципфа (С3),

три подсерии устойчивости при различном уровне шума (С4а, С4б, С4с) и чувствительность к параметрам (С5) (табл. 2).

Для сравнения используются четыре алгоритма: статическое распределение без взвешивания (А0), статическое взвешивание пропорционально емкости (А1), динамическое взвешивание без бюджета [8] (А2), ограничение нагрузки [11] (А3). Каждая серия выполнена в 30 повторениях. Статистическая значимость различий проверяется критерием Вилкоксона при $\alpha = 0,05$. Корректность программной реализации проверялась набором из 57 модульных тестов; на каждом шаге каждого повторения контролировалось выполнение пяти инвариантов.

Результаты исследования и их обсуждение

В стационарном режиме (серия С1) метод снижает средний дисбаланс D на 21,5 % по сравнению с алгоритмом равномерного распределения А0 (0,177 против 0,225, $p < 0,001$). По качеству балансировки метод сопоставим с алгоритмом статического взвешивания А1, использующим априорные знания об емкостях узлов ($D = 0,175$, различие незначимо). Стоимость перестройки составляет $M_cum = 0,063$, тогда как у алгоритма динамического взвешивания А2 $M_cum = 30,028$ при сопоставимом снижении дисбаланса (рис. 2).

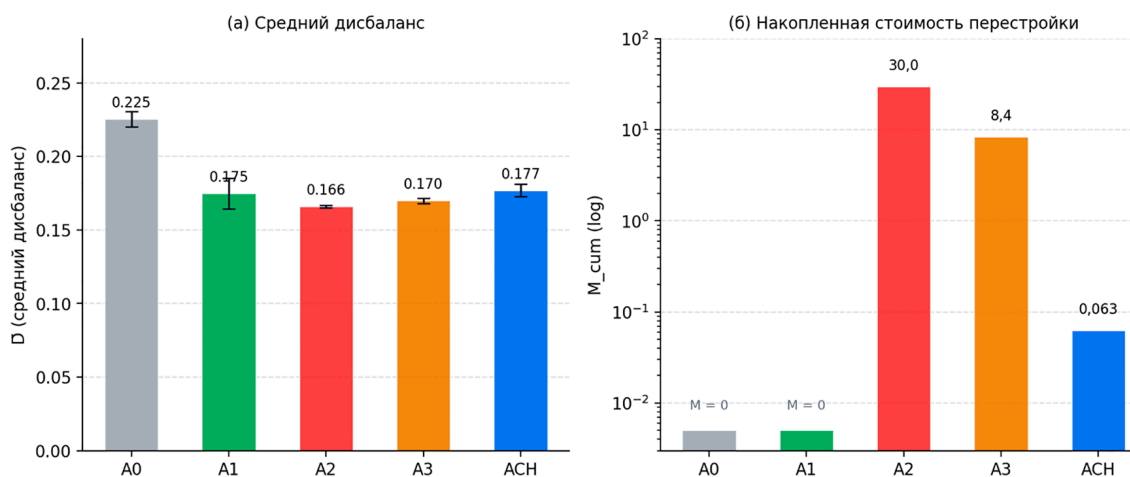


Рис. 2. Сравнение алгоритмов по среднему дисбалансу D (а) и стоимости перестройки M_cum (б) в серии С1
Примечание: составлен авторами по результатам исследования

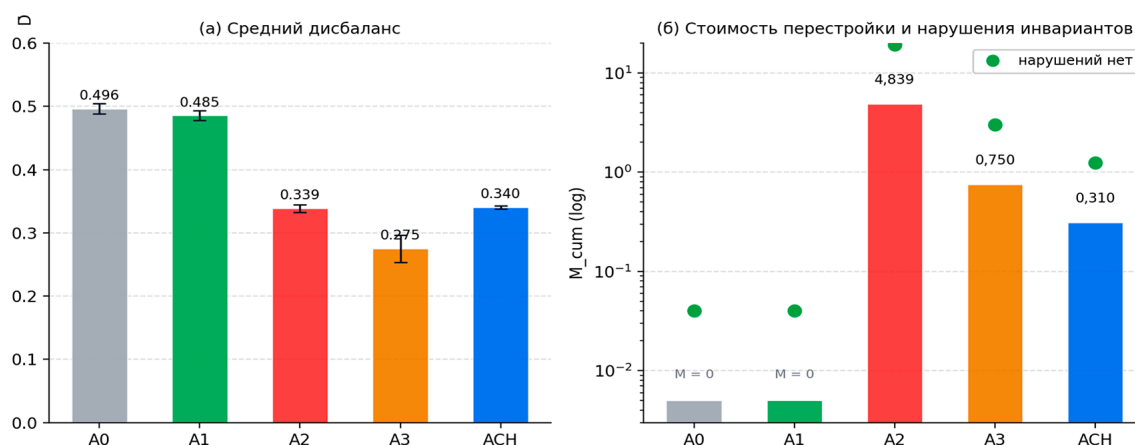


Рис. 3. Сравнение алгоритмов в серии С3: средний дисбаланс D (а) и стоимость перестройки M_cum с числом нарушений инвариантов (б).
Кружки обозначают число нарушений: красный – нарушения, зеленый – ноль
Примечание: составлен авторами по результатам исследования

Таблица 3

Сравнение адаптивных алгоритмов в серии С3 (30 повторений)

Алгоритм	$D \pm CI_{95}$	M_{cum}	π_{chg}	n_{viol}
A0 (статич.)	$0,496 \pm 0,008$	0,000	0,000	0
A1 (взвешен.)	$0,485 \pm 0,008$	0,000	0,000	0
A2 (динамич.)	$0,339 \pm 0,006$	4,839	0,155	0
A3 (ограничен.)	$0,275 \pm 0,023$	0,750	0,045	0
АСН (предлагаемый)	$0,340 \pm 0,003$	0,310	0,058	0

Примечание: CI_{95} – 95 %-ный доверительный интервал, π_{chg} – доля шагов с ненулевой перестройкой, n_{viol} – число нарушений инвариантов.

Составлена авторами на основе полученных данных в ходе исследования

Серия С3 моделирует изменение состава узлов (удаление двух узлов на шаге $t = 150$, добавление трех на шаге $t = 300$) при неравномерном распределении запросов по закону Ципфа с параметром $s = 1,0$. Снижение D в ней составляет 31,6 % относительно A0 (0,340 против 0,496, $p < 0,001$) и 30,0 % относительно A1 (0,485). В данном сценарии метод показывает статистически значимое снижение дисбаланса относительно как A0, так и A1 ($p < 0,001$): статические алгоритмы не адаптируют распределение при изменении состава кластера. Стоимость перестройки АСН составила $M_{cum} = 0,310$ против 4,839 у A2 (табл. 3, рис. 3). При смене состава узлов статические алгоритмы не перераспределяют ключевое пространство без явного пересчета весов; АСН перераспределяет ключевое пространство инкрементально, в рамках бюджетного ограничения.

Отсутствие нарушений у предлагаемого метода объясняется конструкцией: бюджет $m(t)$ вычисляется до начала перестройки,

процедура выбора токенов прекращается при достижении предела; фактический расход дополнительно проверяется инвариантом с дискретной верхней оценкой $M_{keys}(t) \leq m + |O| \cdot \max_s L_s$ [23].

Серии С4а–С4с иллюстрируют изменение характеристик метода при возрастающем уровне искажений телеметрии (рис. 4). При шуме $\sigma = 0,02$ (С4а) и $\sigma = 0,05$ (С4б) метод снижает D на 23–25 % по сравнению с A0 ($p < 0,001$) при качестве балансировки, практически неотличимом от A1. В подсерии С4с ($\sigma = 0,10$, лаг $d = 2$ шага, пропуски $p_{miss} = 10\%$) снижение дисбаланса относительно A0 составляет 9,8 % ($D = 0,198$ против 0,219, $p < 0,001$). Вместе с тем накопленная стоимость перестройки в С4с возрастает до $M_{cum} = 4,565$ (против 0,070 в С4а и 0,167 в С4б): шум телеметрии превышает зону нечувствительности гистерезиса и вызывает избыточные переназначения токенов. Нарушений инвариантов во всех подсериях не зафиксировано.

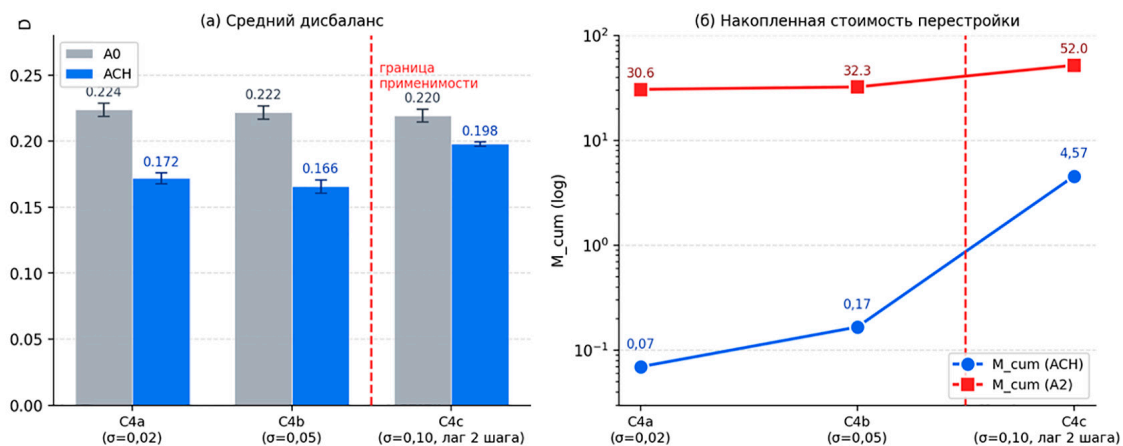


Рис. 4. Градиент деградации АСН при увеличении искажений телеметрии
Серые столбцы – алгоритм статического распределения A0, синие – АСН
Красная штриховая линия обозначает границу применимости ($\sigma > \varepsilon_{off} \wedge d > 1$)
Примечание: составлен авторами по результатам исследования

Таблица 4

Чувствительность АСН к параметрам ε_{on} , κ , τ , M_{max}

Параметр	Значение	D	M_{cum}	π_{chg}	Замечание
ε_{on}	0,05	0,169	0,234	0,256	Агрессивная коррекция
ε_{on}	0,10 (база)	0,176	0,064	0,036	Рабочая точка
ε_{on}	0,15	0,208	0,000	0,000	Коррекция не запускается
κ	0,05–0,20	0,176	0,064	0,036	В диапазоне без эффекта
τ	0,70–0,80	0,225	0,000	0,000	$\gamma = 0$, подавление
τ	0,85 (база)	0,176	0,064	0,036	Возобновление коррекции
M_{max}	0,005–0,10	0,176	0,064	0,036	Ограничен $M_{N/N}$

Примечание: по параметрам κ и M_{max} представлен интервал значений, в котором агрегированные показатели одинаковы в пределах разрешающей способности эксперимента (10 повторений). Полные результаты sweep по 5 значениям ε_{on} , 4 значениям κ , 5 значениям τ и 5 значениям M_{max} сохранены в `repos./results/C5/sensitivity.json`. Доверительные интервалы (95 %) для всех метрик не превышают: $\Delta D < 0,008$, $\Delta M_{cum} < 0,012$, $\Delta \pi_{chg} < 0,015$.

Составлена авторами на основе полученных данных в ходе исследования.

Результат подсерии C4c определяет количественную границу применимости метода: эффективность сохраняется при $\sigma_v \leq \varepsilon_{off}$ (порог выключения коррекции, в эксперименте $\varepsilon_{off} = 0,04$) и при лаге $d \leq 1$ шаг. При нарушении этих условий шум телеметрии превышает зону нечувствительности гистерезиса, что приводит к ложным включениям коррекции и нецелесообразным переназначениям.

Для пары АСН–А0 в каждой серии (30 повторений) рассчитана дельта Клиффа $|\delta|$ как мера эффекта: C1 – 0,996; C2 – 1,000; C3 – 1,000; C4a – 1,000; C4b – 1,000; C4c – 0,936. Значения близки к 1 и указывают на сильное разделение эмпирических распределений D . По шкале Romano и др. (2006) $|\delta| \geq 0,474$ классифицируется как «большой» эффект, что выполнено во всех сериях, включая C4c ($|\delta| = 0,936$). Индивидуальные значения D по 30 повторениям, использованные для расчета $|\delta|$, и позволяют независимо построить графики `boxplot` и `ECDF` распределений D .

По результатам анализа чувствительности (серия C5), поведение метода определяют два параметра: порог включения коррекции ε_{on} и порог подавления перестройки τ . При $\varepsilon_{on} = 0,05$ переназначения выполняются на 25,6 % шагов ($D = 0,169$); при $\varepsilon_{on} \geq 0,15$ перестройка не инициируется ($D \approx 0,208$). Параметр τ демонстрирует пороговое поведение: при $\tau \leq 0,80$ перестройка полностью подавлена ($\gamma = 0$, так как $total_load \geq \tau$), при $\tau \geq 0,85$ перестройка возобновляется. Детальные результаты серии C5 с разбиением по четырем параметрам (ε_{on} , κ , τ , M_{max}) приведены в табл. 4. Диапазоны выбраны симметрично относи-

тельно базовых значений из табл. 1a: $\varepsilon_{on} \in [0,05; 0,20]$ покрывает область от агрессивной до консервативной коррекции; $\kappa \in [0,05; 0,20]$ – от двукратного снижения до удвоения базового значения $\kappa = 0,10$; $\tau \in [0,70; 0,95]$ – от уровня τ_{min} до близко к насыщению; $M_{max} \in [0,005; 0,10]$ – от пятикратного снижения до пятикратного увеличения базового бюджета 0,02. Параметры κ и M_{max} в исследованном диапазоне не показали значимого влияния на агрегированные показатели D и M_{cum} : при базовой удельной потребности $|O| \cdot g_i \approx 0,0025$ на шаг ограничивающим всегда выступает компонент $M_{N/N} = 0,0025$, а не M_{max} ; параметр κ входит как мультипликативное смещение в множитель η и при M_{keys} , ограниченном бюджетом, его варьирование в указанном диапазоне не выходит за погрешность повторений.

При развертывании в производственной среде метод АСН работает как отдельный сервис маршрутизации между клиентами и узлами кластера (рис. 1). Источником телеметрии $m_i(t)$ служит внешняя система мониторинга (Prometheus, Zabbix) или встроенные средства СУБД. Подсистема миграции данных вызывается контуром управления через потоковую передачу, `bulk transfer` или штатные механизмы перебалансировки СУБД; переназначение токена фиксируется только после завершения переноса соответствующего сегмента ключевого пространства. Это позволяет не изменять протоколы транзакционной согласованности и репликации при условии, что переключение маршрутизации выполняется атомарно после подтвержденного переноса сегмента и согласовано с механизмом репликации.

Выводы

1. Предложен метод адаптивного консистентного хеширования, включающий мультипликативный коэффициент интенсивности управления $C(t)$, бюджет переназначений $m(t)$ и гистерезис порогов чувствительности. Ограничения на стоимость перестройки соблюдаются на каждом шаге и на заданном временном горизонте с учетом дискретной верхней оценки перерасхода на длины токенов.

2. Вычислительный эксперимент (7 серий, 30 повторений каждая для C1–C4) показал статистически значимое ($p < 0,001$) снижение дисбаланса нагрузки относительно алгоритма равномерного распределения A0 в диапазоне 9,8–55,4 % в зависимости от сценария. В стационарных условиях качество балансировки сопоставимо с алгоритмом статического взвешивания A1, который требует априорного знания емкостей узлов. При динамическом изменении состава кластера (C3) снижение дисбаланса относительно A0 и A1 статистически значимо ($p < 0,001$). Стоимость перестройки ниже, чем у алгоритма динамического взвешивания A2.

3. Установлены условия ограниченной применимости метода: при одновременном действии шума телеметрии $\sigma_v > \varepsilon_{off}$ и задержки измерений $d \geq 2$ шагов стоимость перестройки возрастает до $M_{sum} = 4,565$, а снижение дисбаланса ограничено 9,8 %; без предварительной фильтрации телеметрии применение метода в таких условиях нецелесообразно. При доминировании горячих ключей (распределение Ципфа, $s = 1,0$) снижение дисбаланса ограничено 31,6 %.

4. Среди возможных направлений дальнейшей работы – распространение подхода на управление отдельными горячими ключами и оперативная адаптация параметров гистерезиса.

Список литературы

1. Karger D., Lehman E., Leighton T., Panigrahy R., Levine M., Lewin D. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the World Wide Web // Proceedings of the 29th Annual ACM Symposium on Theory of Computing. 1997. P. 654–663. DOI: 10.1145/258533.258660.

2. DeCandia G., Hastorun D., Jampani M., Kakulapati G., Lakshman A., Pilchin A., Sivasubramanian S., Vosshall P., Vogels W. Dynamo: Amazon's highly available key-value store // ACM SIGOPS Operating Systems Review. 2007. Vol. 41. Is. 6. P. 205–220. DOI: 10.1145/1323293.1294281.

3. Lakshman A., Malik P. Cassandra: a decentralized structured storage system // ACM SIGOPS Operating Systems Review. 2010. Vol. 44. Is. 2. P. 35–40. DOI: 10.1145/1773912.1773922.

4. Stoica I., Morris R., Liben-Nowell D., Karger D. R., Kaashoek M. F., Dabek F., Balakrishnan H. Chord: a scalable peer-to-peer lookup protocol for internet applications // IEEE/ACM Transactions on Networking. 2003. Vol. 11. Is. 1. P. 17–32. DOI: 10.1109/TNET.2002.808407.

5. Eisenbud D. E., Yi C., Contavalli C., Smith C., Kononov R., Mann-Hielscher E., Cilingiroglu A., Cheyney B., Shang W., Hosein J. D. Maglev: A Fast and Reliable Software Network Load Balancer // Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 2016). 2016. P. 523–535. URL: <https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eisenbud> (дата обращения: 24.04.2026).

6. Mendelson G., Vargaftik S., Barabash K., Lorenz D. H., Keslassy I., Orda A. AnchorHash: A Scalable Consistent Hash // IEEE/ACM Transactions on Networking. 2021. Vol. 29. Is. 2. P. 517–528. DOI: 10.1109/TNET.2020.3039547.

7. Зернов А. С., Ожиганов А. А. Горизонтальное масштабирование базы данных с использованием консистентного хеширования // Известия высших учебных заведений. Приборостроение. 2017. Т. 60. № 3. С. 234–238. URL: https://openbooks.itmo.ru/ru/article/16608/gorizontalnoe_masshtabirovanie_bazy_dannyh_s_ispolzovaniem_konsistentnogo_heshirovaniya.htm (дата обращения: 24.04.2026). DOI: 10.17586/0021-3454-2017-60-3-234-238.

8. Алексеев И. А., Егунов В. А., Паниюлайтис С. В., Чекушкин А. А. Методы и средства балансировки нагрузки в неоднородных вычислительных системах // Инженерный вестник Дона. 2020. № 11. С. 160–168. URL: <https://www.ivdon.ru/ru/magazine/archive/n11y2020/6667> (дата обращения: 24.04.2026).

9. Wang C., Chen Y., Wang J., Wang B. A Dynamic Weights Consistent Hashing Load Balancing Method Based on Heuristic Optimization // 2023 3rd International Conference on Electronic Information Engineering and Computer Science (EIECS). 2023. P. 492–496. DOI: 10.1109/EIECS59936.2023.10435520.

10. Куровский С. В., Мишин Д. А., Шульман В. Д. Алгоритм балансировки нагрузки в гетерогенной среде вычислительной системы // Международный научно-исследовательский журнал. 2025. № 5 (155). URL: <https://research-journal.org/archive/5-155-2025-may/10.60797/IRJ.2025.155.67> (дата обращения: 24.04.2026). DOI: 10.60797/IRJ.2025.155.67. EDN: HKQFAE.

11. Mirrokni V., Thorup M., Zadimoghaddam M. Consistent hashing with bounded loads // Proceedings of the 29th Annual ACM-SIAM Symposium on Discrete Algorithms. 2018. P. 587–604. URL: <https://research.google/pubs/consistent-hashing-with-bounded-loads/> (дата обращения: 24.04.2026). DOI: 10.1137/1.9781611975031.39.

12. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. Sebastopol: O'Reilly Media, 2017. 616 p. [Электронный ресурс]. URL: <https://www.oreilly.com/library/view/designing-data-intensive-applications/9781491903063/> (дата обращения: 24.04.2026). ISBN 978-1-4493-7332-0.

13. Боронников А. С., Цынгалев П. С., Ильин В. Г., Деменкова Т. А. Оценка эффективности балансировщика соединений PgBouncer для оптимизации вычислительных ресурсов реляционных баз данных // Russian Technological Journal. 2024. Т. 12. № 3. С. 7–24. URL: <https://www.rti-journal.ru/jour/article/view/915> (дата обращения: 24.04.2026). DOI: 10.32362/2500-316X-2024-12-3-7-24. EDN: BNQNDI.

14. Levi Y., Keslassy I. Beyond the Ring: Quantized Heterogeneous Consistent Hashing // 2023 IEEE 31st International Conference on Network Protocols (ICNP). 2023. P. 1–12. URL: <https://icnp23.cs.ucr.edu/assets/papers/icnp23-final77.pdf> (дата обращения: 24.04.2026). DOI: 10.1109/ICNP59255.2023.10355577.

15. Coluzzi M., Brocco A., Antonucci A., Leidi T. MementoHash: A Stateful, Minimal Memory, Best Performing Consistent Hash Algorithm // IEEE/ACM Transactions on Networking. 2024. Vol. 32. Is. 4. P. 3528–3543. DOI: 10.1109/TNET.2024.3393476.

16. Dong C., Wang F., Feng D. DxHash: A Memory-saving Consistent Hashing Algorithm // ACM Transactions on Internet Technology. 2024. Vol. 24. Is. 1. Art. 3. P. 1–22. DOI: 10.1145/3631708.

17. Lamping J., Veach E. A Fast, Minimal Memory, Consistent Hash Algorithm // arXiv preprint arXiv:1406.2294. 2014. URL: <https://arxiv.org/abs/1406.2294> (дата обращения: 10.05.2026). DOI: 10.48550/arXiv.1406.2294.
18. Coluzzi M., Brocco A., Leidi T. Consistently Faster: A Survey and Fair Comparison of Consistent Hashing Algorithms // Proceedings of the 31st Symposium on Advanced Database Systems (SEBD 2023). CEUR Workshop Proceedings. 2023. Vol. 3478. P. 51–64. URL: <https://ceur-ws.org/Vol-3478/paper03.pdf> (дата обращения: 10.05.2026).
19. Zhang M. Dynamic Load Balance Strategy Based on Hash Slots in Distributed Storage Environment // 2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC). IEEE, 2022. P. 485–488. URL: <https://ieeexplore.ieee.org/document/10075322> (дата обращения: 10.05.2026). DOI: 10.1109/ICFTIC57696.2022.10075322.
20. Waqas M., Lin S.-J., Liu B., Fazi A. On the Modification of Ring Consistent Hash Uniformity: A Case Study // 2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2024. P. 4778–4783. DOI: 10.1109/SMC54092.2024.10831430.
21. Ji K., Quan G., Tan J. Asymptotic Miss Ratio of LRU Caching with Consistent Hashing // Proceedings of the IEEE INFOCOM 2018 – IEEE Conference on Computer Communications. 2018. P. 450–458. DOI: 10.1109/INFOCOM.2018.8485860.
22. Li J., Nelson J., Michael E., Jin X., Ports D. R. K. Pegasus: Tolerating Skewed Workloads in Distributed Storage with In-Network Coherence Directories // Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation (OSDI '20). USENIX Association, 2020. P. 387–406. URL: <https://www.usenix.org/conference/osdi20/presentation/li-jialin> (дата обращения: 10.05.2026). ISBN 978-1-939133-19-9.
23. Батанов А. О., Литвинова А. С. Передовой подход к распределенным системам: динамические алгоритмы и эффективная обработка больших данных // Современные наукоемкие технологии. 2025. № 3. С. 8–13. URL: <https://top-technologies.ru/ru/article/view?id=40317> (дата обращения: 24.04.2026). DOI: 10.17513/snt.40317. EDN: HRSYLL.

Конфликт интересов: Авторы заявляют об отсутствии конфликта интересов.

Conflict of interest: The authors declare that there is no conflict of interest.

Финансирование: Авторы заявляют об отсутствии внешнего финансирования.

Financing: The research was performed without external funding.