УДК 004.056 Вестник УрФО № 3(57) / 2025, с. 30–39

Мельников Д. Ю.

DOI: 10.14529/secur250304

МОДЕЛЬ ХРАНЕНИЯ ДАННЫХ В NOSQL-СУБД «APACHE CASSANDRA» В ЗАДАЧАХ ОБНАРУЖЕНИЯ КОМПЛЕКСНЫХ КОМПЬЮТЕРНЫХ АТАК

В статье предлагается модель представления данных в NoSQL-СУБД «Apache Cassandra» для системы обнаружения комплексных компьютерных атак на основе анализа действий пользователей, сетевых взаимодействий и сопоставления с базой знаний MITRE ATT&CK. Сформулированы шесть целевых требований к системе хранения данных, такие как: высокая пропускная способность приема событий, предсказуемые ключ-ориентированные чтения, гибкость схемы для полуструктурированных данных, управление жизненным циклом, масштабируемость и совместимость с ML-модулями. На их основе разработана практическая модель с десятью ключевыми таблицами, предложены правила группировки по временным окнам, стратегия оптимизации хранения для временных рядов TWCS и механизмы автоматического удаления устаревших событий по времени жизни TTL для оперативного окна хранения. Приведено теоретическое обоснование проектных решений, выполнены расчеты объемов хранения и размеров партиций, а также построена аддитивная модель end-to-end задержки. Выполнено аналитическое моделирование трех сценариев нагрузки, показаны ориентировочные требования к ресурсам и p50/p95/p99 латентности. Обозначены ограничения моделирования.

Ключевые слова: *Apache Cassandra, NoSQL, UEBA, MITRE ATT&CK, комплексная компьютерная атака.*

DATA STORAGE MODEL IN THE NOSQL DATABASE SYSTEM «APACHE CASSANDRA» FOR DETECTING COMPLEX COMPUTER ATTACKS

The article proposes a model for representing data in the Apache Cassandra NoSQL database management system for a system that detects complex computer attacks based on the analysis of user actions, network interactions, and comparison with the MITRE ATT&CK knowledge base. Six target requirements for the data storage system are formulated, such as high event reception throughput, predictable key-oriented reads, schema flexibility for semi-structured data, lifecycle management, scalability, and compatibility with ML modules. Based on these requirements, a practical model with ten key tables has been developed, rules for grouping by time windows have been proposed, a storage optimization strategy for TWCS time series has been proposed, and mechanisms for automatic deletion of outdated events based on TTL lifetime for the operational storage window have been proposed. A theoretical justification of the design solutions is provided, calculations of storage volumes and partition sizes are performed, and an additive end-to-end delay model is constructed. Analytical modeling of three load scenarios has been performed, and approximate resource requirements and p50/p95/p99 latency have been shown. The limitations of the modeling are indicated.

Keywords: *Apache Cassandra, NoSQL, UEBA, MITRE ATT&CK, complex computer attack.*

Введение

Современные информационные системы все чаще становятся объектами комплексных компьютерных атак [1]. В предыдущей работе [2] автором была предложена модель обнаружения комплексных компьютерных атак (далее – ККА), основанная на построении и динамическом обновлении профилей действий пользователей (далее – UEBA) [3], обнаружении аномалий с помощью метода машинного обучения - автокодировщика и графовых сверточных сетей (далее – GCN), а также сопоставлении последовательностей событий с тактиками и техниками базы MITRE ATT&CK. Модель позволяет выявлять не только единичные аномальные действия, но и объединять их в логически связанные группы, соответствующие этапам ККА. В основе подхода — представление потенциальной комплексной компьютерной атаки как последовательности связанных групп событий, каждая из которых может соответствовать определенной тактике/технике базы знаний MITRE [4]. Данное представление позволяет переводить поток телеметрии в структурированные цепочки и оценивать их согласованность во времени и по контексту. В статье подробно описаны каналы сбора и нормализации событий, логика построения контекстных связей и механизм сопоставления групп событий с базой знаний MITRE ATT&CK

Реализация предлагаемой модели требует надежной и масштабируемой инфраструктуры хранения данных, способной обрабатывать большие объемы разнородной информации в реальном времени. Реляционные системы управления базами данных (далее – СУБД) являются недостаточно гибкими и не справляются с требованиями к скорости записи, масштабируемости и структурной неоднородности данных [5]. Альтернативный подход реализован в нереляционных СУБД

(далее – NoSQL-СУБД), которые обеспечивают гибкость схемы хранения, горизонтальную масштабируемость и устойчивость к нагрузкам.

1. Требования к хранилищу данных

В условиях постоянного роста объемов телеметрии и разнообразия источников, подход к организации хранения данных напрямую влияет на эффективность системы обнаружения. Предлагаемый подход обнаружения ККА, требует одновременной работы с различными типами данных, такими как журналы аутентификации, сетевыми соединениями, данные от средств защиты. Вышеперечисленные данные могут поступать в асинхронном режиме, отличаться по структуре и длительности хранения. Следовательно, хранилище должно не просто выполнять функцию долговременного хранения, но и обеспечивать эффективный доступ, агрегацию и подготовку данных для последующего анализа. Формализуем целевой набор требований к хранилищу данных исходя из предложенной модели обнаружения ККА:

R1 — высокая пропускная способность записи. Ключевым критерием яв-ляется способность хранилища поддерживать устойчивую запись при высоко-интенсивном и неравномерном потоке телеметрии с минимальной задержкой операций записи и отсутствием блокировок, связанных с транзакционными ог-раничениями. Данное свойство критично для предотвращения потери событий и обеспечения непрерывности.

R2 — предсказуемые и быстрые ключ-ориентированные чтения. Для ана-лиза действий пользователей и корреляции аномальных событий необходимы быстрые запросы типа «последние N событий по ключу». Поэтому важным критерием является низкая латентность при доступе к данным по ключевым атрибутам. Это обеспечивает оперативность аналитики и принятия решений в реальном времени.

R3 — гибкость схемы хранения и поддержка полуструктурированных данных. Структура событий может быть изменчива, могут появляться новые поля или меняться формат записи. Поэтому возможность хранения полуструктурированных данных и добавления новых атрибутов без полной миграции схемы является необходимым критерием.

R4 — управление жизненным циклом данных. Для снижения объема хра-нилища и устранения шума необходимо наличие встроенных механизмов автоматического удаления устаревших событий по времени жизни (Time To Live, далее — TTL) [6]. Критерием является поддержка политики выборочного хранения, а также возможность архивации и изоляции событий.

R5 — горизонтальная масштабируемость и отказоустойчивость. Хранилище должно линейно масштабироваться при росте объема данных и обеспечивать круглосуточную доступность. Возможность расширения кластера за счет добавления узлов без остановки сервиса, наличие встроенной репликации и балансировки нагрузки, а также устойчивость к сбоям отдельных узлов являются важными факторами. Это гарантирует стабильность системы даже при увеличении нагрузки и в условиях отказов оборудования.

R6 — совместимость с внешними аналитическими и модулями машинного обучения (далее – ML модули). Способность хранилища предоставлять данные для аналитики и машинного обучения. Важными свойствами являются: эффективная выборка признаков по ключам, хранение векторных представлений объектов (далее – эмбеддинги), формируемых ML-модулями, совместно с исходными событиями, поддержка агрегированных профилей пользователей и экспорт данных в форматы, удобные для обучения моделей.

Таким образом, каждое из требований к хранилищу (R1–R6) соответствует функциональным и архитектурным особенностей предложенной модели и определяет необходимость использования специализированного подхода к проектированию хранилища данных для задач обнаружения ККА. Все перечисленные характеристики в наибольшей степени реализуются в NoSQL-СУБД [4, 5], что и определяет выбор типа СУБД в качестве технологической основы.

2. Выбор NoSQL - СУБД и обоснование

Для выбора NoSQL-СУБД был проведен сравнительный анализ распро-страненных решений, на основе научно-технической литературы [5,7,8,9] и критериев, сформулированных ранее. Оценка производилась по пятибалльной шкале, однако в ходе анализа значения 1 и 2 не использовались, поскольку все рассматриваемые NoSQL-СУБД демонстрируют удовлетворительный уровень соот-

Таблица 1

Сравнительный анализ NoSQL-СУБД

СУБД	R1 (0.20)	R2 (0.15)	R3 (0.15)	R4 (0.15)	R5 (0.20)	R6 (0.15)	Итоговая оценка
Cassandra	5	5	4	5	5	4	4,7
HBase	4	4	3	4	4	3	3,7
Elasticsearch	3	3	4	4	3	5	3,8
MongoDB	3	4	5	3	3	4	3,8

ветствия требованиям. Итоговый балл рассчитан как средневзвешенное значение с учетом весов критериев. Весовые коэффициенты определялись автором на основе приоритета каждого критерия для системы хранения данных.

По итогам сравнительного анализа, можно сделать вывод о том, что NoSQL-СУБД «Apache Cassandra» демонстрирует наиболее высокую степень соответствия целевым требованиям, особенно по приоритетным критериям R1 R2, что позволяет рассматривать ее в качестве приоритетной платформы для построения системы обнаружения ККА.

3. Описание архитектуры модели хранения данных

Предлагаемая архитектура модели хранения данных (Рисунок 1) разрабо-тана с учетом обозначенных требований и реализует принцип запрос-ориентированного проектирования (query-driven design), при котором схема хранения определяется набором типовых запросов системы.

Проектирование модели хранения данных основано на ключевых принципах: денормализации и дублировании данных в соответствии со сценариями доступа для мини-

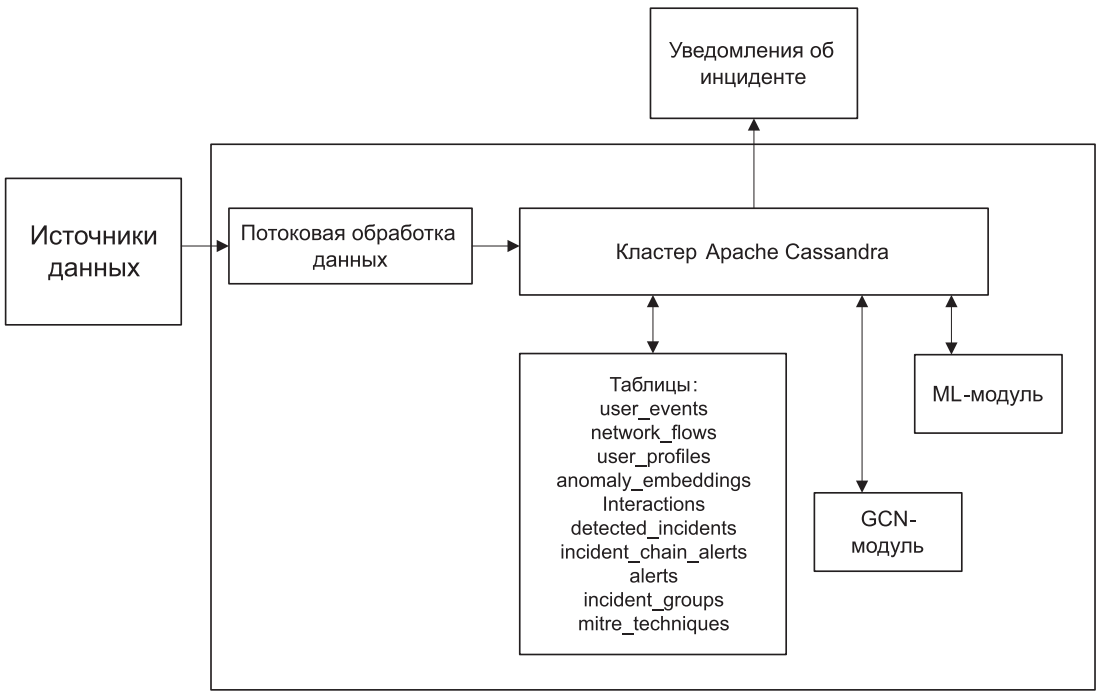

Рис. 1. Архитектура модели хранения данных в NoSQL-СУБД «Apache Cassandra»

мизации задержек выборки, применении группировки по временным окнам (далее – временная сегментация) для предотвращения неравномерной нагрузки и учете жизненного цикла данных через TTL и стратегии оптимизации хранения для временных рядов (Time Window Compaction Strategy, далее - TWCS) [10,11].

Предложенная архитектура охватывает полный цикл анализа событий от приема необработанных данных до генерации итоговых уведомлений. Ключевые сущности системы, представленные в Таблице 2, спроектированы на осно-ве характерных шаблонов доступа и требований аналитических запросов.

Поток данных в архитектуре реализован следующим образом (Рисунок 2):

1. Данные поступают и сохраняются в таблицах «user_events» и «network_flows».

2. На их основе формируются агрегаты и профили и записываются в «user_profiles», а также вспомогательные представления для расследования.

3. ML-модуль извлекает признаки из «user_events»/ «user_profiles» и вы-числяет эмбеддинги. Эмбеддинги сохраняются в «anomaly_embeddings».

4. Информация о последовательностях взаимодействий и сессиях строится в «interactions», это позволяет восстановить цепочки действий.

5. ML-модуль на основе выявленных аномалий формирует предваритель-ные оповещения и записывает их в таблицу «alerts».

6. Оповещения логически связываются в упорядоченные цепочки в «incident_chain_ alerts».

7. На основе цепочек оповещений формируются окончательные записи инцидентов в «detected_incidents».

8. Инциденты, обладающие сходными признаками и относящиеся к одной ККА, объединяются в таблице «incident_groups».

9. На каждом этапе события и инциденты сопоставляются с тактиками и техниками из базы знаний MITRE ATT&CK, обеспечивая классификацию и добавление атрибутов для анализа.

Таблица 2

Сводная таблица ключевых сущностей

Таблица	Описание	Ключи (Partition Key, Clustering Key)
user_events	Информация о действиях пользователей	PK: (event_type, event_date); CK: event_time
network_flows	Сетевые соединения	PK: (user_id, event_date) CK: event_time
user_profiles	Профили действий пользователей	PK: user_id
anomaly_embeddings	Эмбеддинги событий, полученные с помощью моделей машинного обучения	PK: (user_id, event_date); CK: event_time
interactions	Взаимодействия между субъектами и объектами	PK: (user_id, interaction_date); CK: time
detected_incidents	Обнаруженные инциденты (аномалии)	PK: (user_id, detected_date); CK: time
incident_chain_alerts	Агрегированные инциденты Цепочки инцидентов	PK: (alert_date); CK: (time, alert_id)
alerts	Финальные оповещения	PK: (created_date); CK: (time, alert_id)
incident_groups	Объединяет инциденты в логические группы с агрегированной оценкой	PK: (group_id) CK:group_date
mitre_techniques	Справочная таблица с описаниями техник и тактик базы знаний MITRE ATT&CK	PK: technique_id

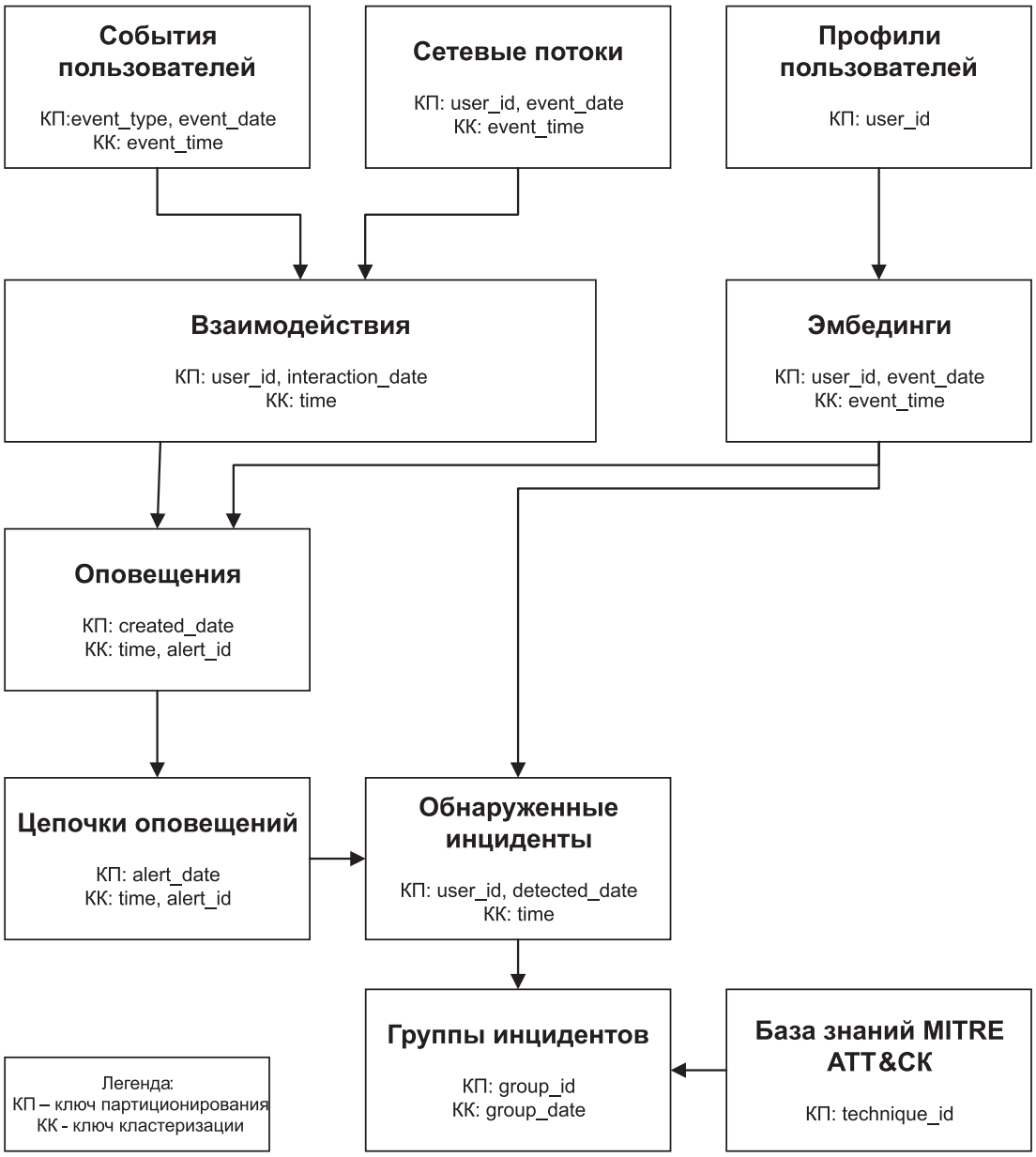

Рис. 2. Модель представления данных в NoSQL-СУБД «Apache Cassandra»

Использование временной сегментации, встроенных TTL, а также ориен-тированной на запросы схемы позволяет обеспечить стабильную работу систе-мы при высоких нагрузках. Вся архитектура организована таким образом, чтобы ускорить работу аналитических модулей и обеспечить устойчивость к сбоям.

Предложенная архитектура модели хранения данных в NoSQL-СУБД «Apache Cassandra» обеспечивает выполнение целевых требований R1–R6. За-пись телеметрии (R1) реализуется за счет оптимизированных таблиц «user_events» и «network_flows» с применением TTL и стратегий слияния дан-ных, оптимизированных для временных рядов [12]. Предсказуемые и быстрые ключ-запросы (R2) поддерживаются через специализированные таблицы «user_profiles», «interactions» и «alerts». Гибкость в работе с полуструктурированными событиями (R3) достигается использованием денормализации и хранения вложенных атрибутов. Управление жизненным циклом данных (R4) реализуется посредством TTL и дифференцированных сроков хранения для событий, уведомлений и инцидентов. Горизонтальная масштабируемость и

Таблица 3

Основные параметры моделирования

Параметр	Значение
Число узлов в кластере (N)	3
Число реплик фрагмента данных (RF)	3
Средний размер события (s)	500 байт
Сценарии нагрузки (ε)	5 000 / 20 000 / 100 000 событий/в секунду
Максимально допустимый размер партиции	100 MB
TTL для events	7 дней
Compaction strategy	TWCS
Уровень согласованности при записи	ONE
Уровень согласованности при оповещении	QUORUM

отказоустойчивость (R5) обеспечиваются встроенными средствами NoSQL-СУБД «Apache Cassandra», а именно распределенным партиционированием, репликацией и балансировкой нагрузки. Интеграция с аналитическими модулями (R6) достигается через хранение агрегатов в «user_profiles», векторных представлений в «anomaly_ embeddings» и связи с базой знаний MITRE ATT&CK в «mitre_techniques».

4. Экспериментальная оценка и моделирование

Для оценки производительности предложенной модели хранения данных при различных нагрузках необходимо:

• произвести аналитическое моделирование объема хранимых данных и нагрузку на вычислительные ресурсы в зависимости от сценариев нагрузки;

• рассчитать задержку записи и чтения (далее – латентность) в условиях нагрузки (p50/p95/p99);

• оценить совокупную задержку поступления события до генерации соответствующего оповещения (далее - end-to-end задержка).

В качестве методики использованы аналитические расчеты и приближен-ное моделирование на основе характерных параметров NoSQL-СУБД «Apache Cassandra». При моделировании использовались опубликованные результаты тестов производительности и официальная документация, применявшиеся для ориентировочной калибровки параметров модели и оценки компонент латент-ности [10,13,14].

Для оценки эффективности предложенной модели хранения данных была смоделирована работа кластера «Apache Cassandra» с заданными характеристиками. В таблице 3 представлены ключевые параметры, которые использовались в ходе экспериментов.

5. Моделирование объема хранения и потребления ресурсов

Из приведенных данных видно, что объем требуемого хранилища линейно зависит от интенсивности входного потока событий. При росте нагрузки от 5 000 до 100 000 событий/ в секунду суточный объем данных увеличивается почти на порядок. Эти оценки позволяют заранее определить минимально необходимый объем дискового пространства и требования к TTL.

Таблица 4

Результаты моделирования объема хранения данных

Нагрузка (событий/с)	Запись на узел (MB/s)	Суточный объем на узел (ГБ)	Объем (7 дн) на узел (ГБ)
5 000	2,5	216	1 512
20 000	10	864	6 048
100 000	50	4 320	30 240

Таблица 5

Результаты моделирования латентности

Сценарий нагрузки (событий/с)	Запись p50/p95/p99 (мс)	Чтение p50/p95/p99 (мс)
5 000	03.10.1930	04.08.2015
20 000	5 / 25 / 80	6 / 20 / 30
100 000	12 / 120 / 400	15 / 150 / 300

6. Моделирование латентности

Для оценки латентности запросов применен упрощенный метод, основан-ный на опубликованных экспериментальных результатах и их линейной экст-раполяции на более высокие нагрузки. Зависимость времени отклика от интен-сивности входного потока для каждой функциональной компоненты аппроксимировалась посредством масштабирования базового значения с учетом эмпирических коэффициентов чувствительности к росту нагрузки.

Результаты моделирования задержек показывают предсказуемый рост ла-тентности при увеличении интенсивности записи. Для сценариев средней на-грузки система сохраняет p95-латентность в пределах десятков миллисекунд, что подтверждает возможность работы в режиме реального времени. При экс-тремальной нагрузке наблюдается рост задержек, однако система продолжает обеспечивать приемлемую доступность и предсказуемость откликов.

7. Моделирование end-to-end задержки

Дополнительно к оценкам объема хранения и базовых задержек операций была рассчитана end-to-end задержка (таблица 6). В отличие от отдельных метрик записи и чтения, данная характеристика учитывает полный путь события в системе: сохранение в хранилище, формирование агрегированных профилей, обработку методами анализа аномалий и генерацию уведомлений.

Как видно из таблицы, при низкой и средней нагрузке медианная задержка (p50) остается в пределах десятков миллисекунд, а 95 % событий обрабатываются быстрее 350 мс. При высокой нагрузке наблюдается рост задержек, однако даже в этом случае система обеспечивает end-to-end обработку в пределах секунды. Таким образом, предложенная модель хранения данных демонстрирует устойчивость и предсказуемость временных характеристик в условиях различной интенсивности телеметрии.

Аналитическое моделирование показывает, что предложенная модель хранения данных обеспечивает ожидаемую способность выдерживать высокие нагрузки записи и поддерживать низкие p95-латентности чтений при корректном подборе параметров временной сегментации и TTL. Приведенные оценки являются репрезентативными и требуют практического подтверждения. Использование временной сегментации позволяет ограничивать размер каждой партиции и снижает риск образования неравномерно нагруженных партиций, что обеспечивает стабильное время доступа к недавним событиям.

Таблица 6

Результаты моделирования end-to-end задержки

Сценарий нагрузки (ε, событий/с)	E2E p50 (мс)	E2E p95 (мс)
5 000	~35	~120
20 000	~60	~350
100 000	~200	~800+

Заключение

В статье предложена и обоснована модель представления данных для NoSQL-СУБД «Apache Cassandra», ориентированная на задачу обнаружения комплексных компьютерных атак с учетом анализа действий пользователей, сетевых взаимодействий и сопоставления с базой знаний MITRE ATT&CK. В основе работы лежит формализованный набор требований (R1–R6), определяющих ключевые характеристики системы в условиях высоких нагрузок. На основе этих требований была разработана детализированная модель данных с обоснованием выбора схемы временной сегментации и ключей кластеризации, а также стратегий денормализации.

Проведенный анализ демонстрирует, что предложенная архитектура соответствует сформулированным требованиям. Временная сегментация, контроль размеров партиций, политика TTL и стратегия TWCS обеспечивают приемлемые значения объема хранения и латентности при заданных сценариях. Модель демонстрирует устойчивость к росту объема входных данных за счет распределенного хранения и репликации; при этом сохранение прогнозируемых p50/p95-значений достигается путем ограничения размера партиции, подбора числа реплик фрагмента данных и выбора уровней согласованности. Оценки, полученные в ходе моделирования объема хранения и латентности, подтверждают практическую реализуемость подхода при оговоренных допущениях.

Вместе с тем, необходимо отметить ограничения исследования. Численные результаты получены методом аналитического моделирования и синтетической симуляции. Оценка качества ML-модулей требует отдельного исследования на реальных данных, так как текущие ML-результаты носят иллюстративный характер и служат для демонстрации влияния дизайна хранилища на ML-производительность.

Литература

1. IBM. «What is Security Information and Event Management (SIEM)?» // IBM Think, 23 June 2023. – URL: https://www.ibm.com/topics/siem (дата обращения: 20.08.2025).

2. D. Melnikov, «A Model for Detecting Complex Computer Attacks Based on the Analysis of User Actions, Network Interactions and Comparison with the MITRE ATT&CK Knowledge Base» 2025 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russian Federation, 2025, pp. 221-224, doi: 10.1109/USBEREIT65494.2025.11054096.

3. Fortinet. «What is User and Entity Behavior Analytics (UEBA)?» // Fortinet CyberGlossary. – 2024. – URL: https://www.fortinet.com/resources/cyberglossary/what-is-ueba (дата обращения: 20.08.2025).

4. MITRE. «ATT&CK® Design and Philosophy». – March 2020. – URL: https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf (дата обращения: 20.08.2025).

5. Sadalage P. J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. – Addison-Wesley Professional, 2012. – 192 с. – ISBN 978-0321826626.

6. DataStax Docs. «Expiring data with time-to-live (TTL)». – 2022. – URL: https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useExpire.html (дата обращения: 20.08.2025).

7. Chebotko A., Kashlev A., Lu S. «A Big Data Modeling Methodology for Apache Cassandra» // 2015 IEEE International Congress on Big Data (BigData Congress). – 2015. – С. 238-245. – DOI: 10.1109/BigDataCongress.2015.41.

8. Apache Cassandra Documentation. «Evaluating and Refining Data Models». – 2024. – URL: https://cassandra.apache.org/doc/latest/cassandra/data_modeling/evaluating.html (да-та обращения: 20.08.2025).

9. Lakshman A., Malik P. Cassandra: a decentralized structured storage system // ACM SIGOPS Operating Systems Review. – 2010. – Т. 44, № 2. – С. 35-40. – DOI: 10.1145/1773912.1773922.

10. Apache Cassandra Documentation. «Time Window Compaction Strategy (TWCS)». – 2024. – URL: https://cassandra.apache.org/doc/latest/cassandra/operating/compaction/twcs.html (дата обращения: 20.08.2025).

11. DataStax Tutorial. «Getting Started with Time Series Data Modeling». – 2016. – URL: https://docs.datastax.com/en/tutorials/Time_Series.pdf (дата обраще-ния: 20.08.2025).

12. Ellis J. «Apache Cassandra Performance Benchmarking (4.0 brings new collectors) » // DataStax Blog, Apr. 2021. – URL: https://www.datastax.com/blog/apache-cassandra-benchmarking-40-brings-heat-new-garbage-collectors-zgc-and-shenandoah (дата обращения: 20.08.2025).

13. Instaclustr. «Apache Cassandra Data Partitioning». – 2022. – URL: https://www.instaclustr.com/blog/cassandra-data-partitioning/ (дата обращения: 20.08.2025).

14. O'Neil P., Cheng E., Gawlick D., O'Neil E. «The Log-Structured Merge-Tree (LSM-Tree) » // Acta Informatica. – 1996. – Т. 33, № 4. – С. 351-385. – DOI: 10.1007/s002360050048.

References

1. IBM. «What is Security Information and Event Management (SIEM)?» // IBM Think, 23 June 2023. URL: https://www.ibm.com/topics/siem (дата обращения: 20.08.2025).

2. D. Melnikov, «A Model for Detecting Complex Computer Attacks Based on the Analysis of User Actions, Network Interactions and Comparison with the MITRE ATT&CK Knowledge Base» 2025 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT), Yekaterinburg, Russian Federation, 2025, pp. 221-224, doi: 10.1109/USBEREIT65494.2025.11054096.

3. Fortinet. «What is User and Entity Behavior Analytics (UEBA)?» // Fortinet CyberGlossary, 2024. URL: https://www.fortinet.com/resources/cyberglossary/what-is-ueba (accessed: 20.08.2025).

4. MITRE. «ATT&CK® Design and Philosophy», March 2020. URL: https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf (accessed: 20.08.2025).

5. Sadalage P. J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. – Addison-Wesley Professional, 2012. – 192 с. – ISBN 978-0321826626.

6. DataStax Docs. «Expiring data with time-to-live (TTL)», 2022. URL: https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useExpire.html (accessed: 20.08.2025).

7. Chebotko A., Kashlev A., Lu S. «A Big Data Modeling Methodology for Apache Cassandra» // 2015 IEEE International Congress on Big Data (BigData Congress). – 2015. – С. 238-245. – DOI: 10.1109/BigDataCongress.2015.41.

8. Apache Cassandra Documentation. «Evaluating and Refining Data Models», 2024. URL: https://cassandra.apache.org/doc/latest/cassandra/data_modeling/evaluating.html (ac-cessed: 20.08.2025).

9. Lakshman A., Malik P. Cassandra: a decentralized structured storage system // ACM SIGOPS Operating Systems Review. –2010. – Т. 44, № 2. – С. 35-40. – DOI: 10.1145/1773912.1773922.

10. Apache Cassandra Documentation. «Time Window Compaction Strategy (TWCS)», 2024. URL: https://cassandra.apache.org/doc/latest/cassandra/operating/compaction/twcs.html (accessed: 20.08.2025).

11. DataStax Tutorial. «Getting Started with Time Series Data Modeling», 2016. URL: https://docs.datastax.com/en/tutorials/Time_Series.pdf (accessed: 20.08.2025).

12. Ellis J. «Apache Cassandra Performance Benchmarking (4.0 brings new collectors) » // DataStax Blog, Apr. 2021. URL: https://www.datastax.com/blog/apache-cassandra-benchmarking-40-brings-heat-new-garbage-collectors-zgc-and-shenandoah (accessed: 20.08.2025).

13. Instaclustr. «Apache Cassandra Data Partitioning», 2022. URL: https://www.instaclustr.com/blog/cassandra-data-partitioning/ (accessed: 20.08.2025).

14. O'Neil P., Cheng E., Gawlick D., O'Neil E. «The Log-Structured Merge-Tree (LSM-Tree) » // Acta Informatica. – 1996. – Т. 33, № 4. – С. 351-385. – DOI: 10.1007/s002360050048.

МЕЛЬНИКОВ Дмитрий Юрьевич, аспирант Учебно-научного центра «Информационная безопасность» федерального государственного автономного образовательного учреждения высшего образования «Уральский федеральный университет им. первого Президента России Б.Н. Ельцина». 620002, г. Екатеринбург, ул. Мира, 32. E-mail: melnikovdima517@yandex.ru

MELNIKOV Dmitriy Yuryevich, graduate student at the Educational and Scientific Center «Information Security» of the Federal State Autonomous Educa-tional Institution of Higher Education «Ural Federal University named after the first President of Russia B.N. Yeltsin». 620002, Yekaterinburg, st. Mira, 32. E-mail: mel-nikovdima517@yandex.ru