

# ОЦЕНКА ВЛИЯНИЯ ПРОИЗВОДИТЕЛЬНОСТИ СУБД НА МОНИТОРИНГ СОБЫТИЙНЫХ ДАННЫХ

## ESTIMATING THE IMPACT OF DBMS PERFORMANCE ON EVENT DATA MONITORING

**B. Goryachkin  
M. Panov**

*Summary.* Designing a high-load system that generates a large amount of event data, it is necessary to select a DBMS that shows the highest performance in the context of the intended form of working with this data.

We studied the performance of PostgreSQL and MongoDB DBMS based on the specified criteria and parameters. We carried out comparative analysis of the experimental results, and, based on these results, highlight the specifics of using both DBMS in information systems.

Understanding the performance characteristics of DBMS in different information systems with various significant features allows for correctly designed information system, namely, correct selection of a technological solution for storing event data.

*Keywords:* database, DBMS, MongoDB, PostgreSQL, event data, monitoring.

**Горячкин Борис Сергеевич**

кандидат технических наук, доцент;

Московский государственный

технический университет имени Н.Э. Баумана

*bsgor@mail.ru*

**Панов Максим Константинович**

Московский государственный

технический университет имени Н.Э. Баумана

*panovmk@student.bmstu.ru*

*Аннотация.* При проектировании высоконагруженной системы, генерирующей в большом количестве событийные данные, необходимо выбрать СУБД, показывающую наибольшую производительность в контексте предполагаемой формы работы с этими данными.

Рассмотрена производительность СУБД PostgreSQL и MongoDB на основе сформированных критериев и параметров. Проведен сравнительный анализ результатов эксперимента, а также, на их основе, выделена специфика применения обеих СУБД в информационных системах.

Понимание характеристик производительности СУБД в информационных системах с различными особенностями, позволяет корректно произвести проектирование информационной системы, а именно, корректно выбрать технологическое решение для хранения событийных данных.

*Ключевые слова:* база данных, СУБД, MongoDB, PostgreSQL, событийные данные, мониторинг.

### Введение

В условиях стремительного роста объемов данных и сложности современных информационных систем вопрос обработки событийных данных приобретает особую актуальность. Событийные данные, фиксирующие изменения состояния системы или взаимодействия её компонентов, играют ключевую роль в мониторинге процессов и выявлении аномалий. Это особенно важно для критически значимых отраслей, таких как финансы, телекоммуникации, кибербезопасность и промышленность, где задержки или ошибки обработки данных могут привести к значительным убыткам или угрозам безопасности.

Современные системы мониторинга должны обеспечивать обработку больших объемов данных в реальном времени, включая запись, фильтрацию, агрегацию и временной анализ событий. Производительность систем управления базами данных в данном контексте оказывает решающее влияние на эффективность мониторинга. Выбор СУБД зависит от множества факторов, включая характер и объем данных, частоту аналитических запросов, а также требуемую скорость обработки.

Особенности событийных данных, такие как их временная упорядоченность, высокая частота генерации и необходимость обработки в реальном времени, формируют условия, напрямую влияющие на проектирование части системы, отвечающей за их хранение и обработку. Это делает выбор между реляционными и нереляционными СУБД критически важным этапом разработки высокопроизводительных систем мониторинга. В ряде случаев использование подходящей СУБД может снизить задержки, улучшить масштабируемость и обеспечить соответствие системе требованиям высокой отказоустойчивости.

В данной работе проводится сравнение производительности SQL и NoSQL СУБД на примере MongoDB и PostgreSQL, в целях выявления наиболее подходящего решения для проблемы мониторинга событийных данных.

### Предметная область и характеристики экспериментальной среды

Для проведения экспериментального исследования был использован реализованный на языке программирования python генератор событийных данных со сле-

дующими сущностями и их полями (представлена диаграмма сущность-связь для SQL базы данных):

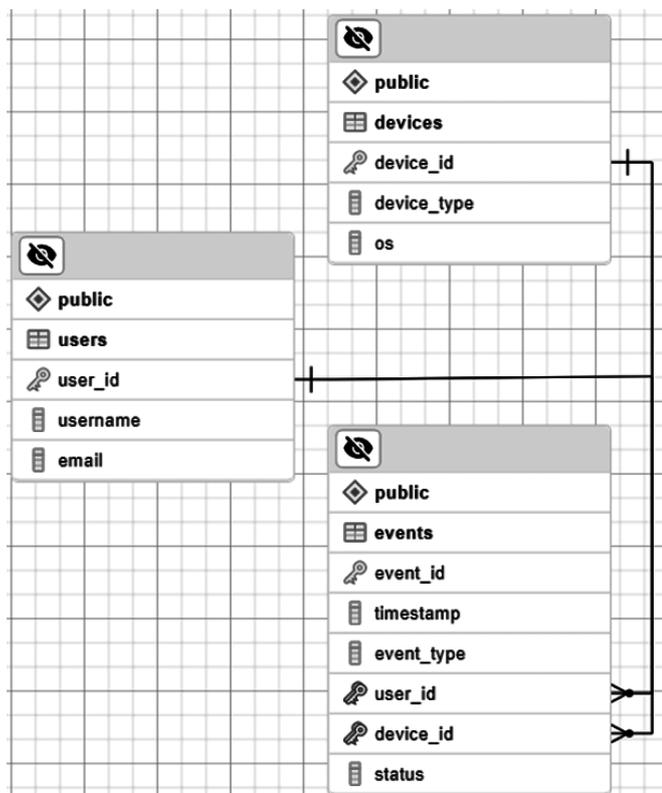


Рис. 1. Структура сгенерированных событий

Для проведения экспериментального исследования, была сформирована рабочая среда на базе операционной системы Windows 11 (для каждой БД в отдельности):

- Характеристики тестового стенда:
  - Процессор: 11th Gen Intel(R) Core (TM) i9-11900K @ 3.50GHz, 3504 МГц, ядер: 8, логических процессоров: 16
  - Полный объем физической памяти 31,8 ГБ
  - SSD накопитель (рис. 2)

| CrystalDiskMark 8.0.5 x64 [Admin]                 |             |      |                          |
|---|-------------|------|--------------------------|
| Файл Настройки Профиль Вид Справка Язык(Language) |             |      |                          |
| All   | 5           | 1GiB | C: 39% (329/853GiB) MB/s |
|   | Read (MB/s) |      | Write (MB/s)             |
| SEQ1M Q8T1  | 3396.55     |      | 3086.22                  |
| SEQ1M Q1T1  | 2473.35     |      | 2846.43                  |
| RND4K Q32T1                                       | 624.34      |      | 360.16                   |
| RND4K Q1T1  | 52.33       |      | 179.53                   |

Рис. 2. Характеристика SSD накопителя

- Версии программного обеспечения:
  - MongoDB Community Server 8.0.3
  - PostgreSQL Server 15

Все эксперименты проводятся на блоках в 1000, 10000, 50000, 100000, 500000 и 1000000 событий.

### Определение критичных критериев и параметров исследования

Событийные данные отражают изменения, происходящие в системе при взаимодействии ее компонентов друг с другом или с пользователем. В основном они фиксируют действия, состоящие из времени события, типа действия, источника и, дополнительных специфических для системы метаданных [4].

Одной из основных характеристик событийных данных является высокий объем и частота их записи. События часто генерируются в больших объемах, особенно в высоконагруженных системах, таких как, распределенные телекоммуникационные системы, социальные сети. Количество событий, генерирующихся в таких системах, может достигать миллионов событий в секунду. Для их обработки часто применяют архитектуры потоковой обработки данных и масштабируемые/отказоустойчивые системы хранения данных [3].

Событийные данные также четко упорядочены во времени. Любое событие содержит временную метку, отражающую момент его возникновения. Это критично для правильной интерпретации данных и своевременного реагирования на аномалии [3].

Не менее важной характеристикой событийных данных является также необходимость их обработки в реальном времени. Это объясняется тем, что событийные данные, по своей сути, предоставляют наиболее актуальное представление о состоянии процессов и компонентов системы, из чего вытекает также их важность для построения мониторинга. Из этого следует необходимость минимизации времени вычисления аналитических метрик из данных.

Также, зачастую событийные данные могут иметь низкую согласованность и переменчивую структуру. Однако, в действительности, это ограничение зависит исключительно от архитектуры системы, генерирующей события. В данном исследовании предполагается постоянная структура обрабатываемых событий.

Опираясь на данные характеристики событийных данных, можно выделить несколько критериев сравнения результатов операций с данными в БД с точки зрения производительности:

- Производительность и масштабируемость записи и чтения блоков данных

- Производительность и масштабируемость обработки данных с привязкой ко времени
- Производительность и масштабируемость аналитических запросов

Производительность СУБД в контексте конкретного критерия возможно оценить исходя из времени, затраченного на выполнение операций, задействованных для выполнения соответствующей критерию задачи.

На основе описанных критериев, с учетом особенностей событийных данных и определения производительности СУБД, представляется возможным выделить следующие **параметры**:

- Время записи блока данных
- Время чтения блока данных
- Время выполнения запроса обработки временного окна блока данных
- Время выполнения агрегационных запросов на блоке данных
- Время выполнения фильтрующих запросов на блоке данных

**Экспериментальная часть**

Для вычисления значений по параметру «время записи блока данных» выполняется запись по всем полям события блока данных в таблице/коллекции (рис. 3, табл. 1). Запись выполнялась через стандартные механизмы вставки: SQL-оператор INSERT INTO для PostgreSQL и метод insertMany () для MongoDB.

MongoDB показала явное преимущество при увеличении объема записываемых данных. При объеме от 100,000 записей её производительность значительно опережала PostgreSQL, что объясняется особенностями

Таблица 1.

Зависимость времени выполнения записи от размера блока данных и СУБД

| № записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 74                                | 70                             |
| 10000             | 694                               | 232                            |
| 50000             | 4016                              | 1073                           |
| 100000            | 9737                              | 2212                           |
| 500000            | 47633                             | 11821                          |
| 1000000           | 102065                            | 22692                          |

Таблица 2.

Зависимость времени выполнения чтения от размера блока данных и СУБД

| № записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 35                                | 2                              |
| 10000             | 51                                | 2                              |
| 50000             | 117                               | 3                              |
| 100000            | 198                               | 3                              |
| 500000            | 898                               | 3                              |
| 1000000           | 1931                              | 3                              |

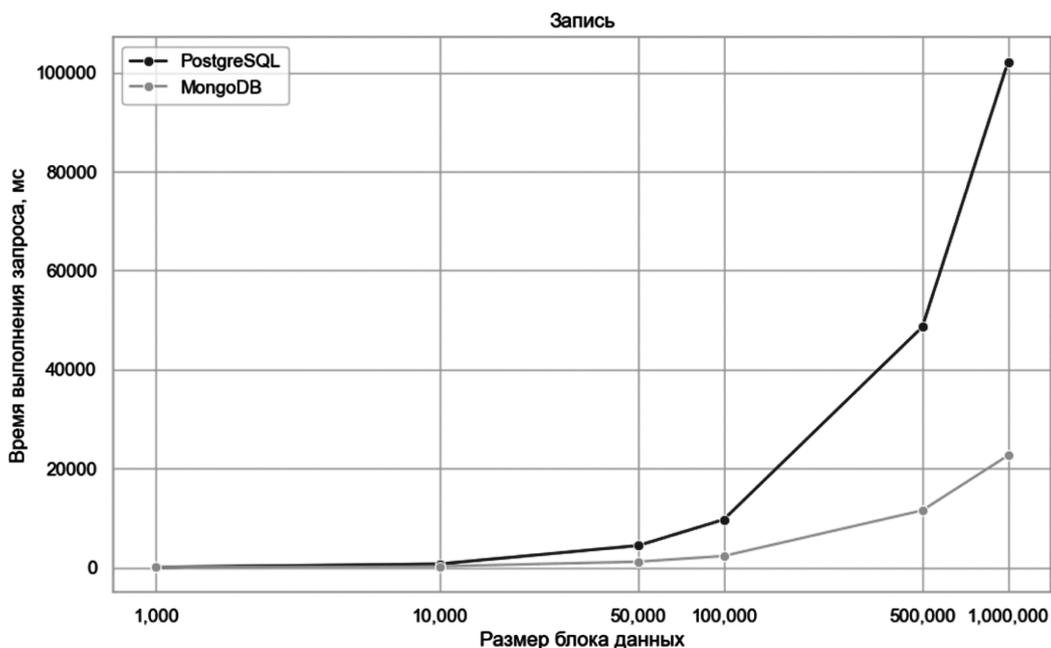


Рис. 3. Зависимость времени выполнения записи от размера блока данных и СУБД

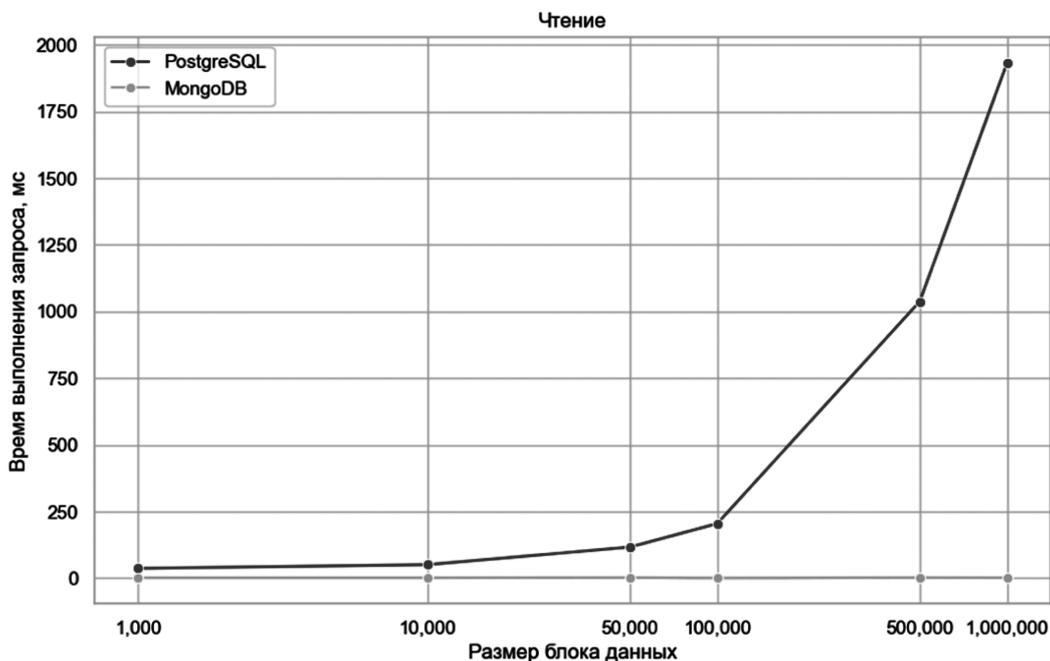


Рис. 4. Зависимость времени выполнения чтения от размера блока данных и СУБД

документной модели, отсутствием строгой схемы и внутренними механизмами оптимизации записи.

Для вычисления значений по параметру «время чтения блока данных» выполняется чтение по всем полям события блока данных в таблице/коллекции (рис. 4, табл. 2). В PostgreSQL использовался SQL-запрос SELECT\*, в MongoDB — метод find ().

При выполнении чтения обе СУБД имеют высокие показатели по времени чтения. Однако, как видно из результатов, чтение в MongoDB масштабируется лучше.

Наиболее наглядно это наблюдается на более крупных блоках данных: при увеличении числа записей в блоке, время практически не изменяется.

Для вычисления значений по параметру «время выполнения запроса обработки временного окна блока данных», были сформированы два запроса:

- «Получить количество событий со статусом failed, полученных с устройства с ОС MacOS за последние 24 часа» (Запрос по временному окну 1, рис. 5, табл. 3)

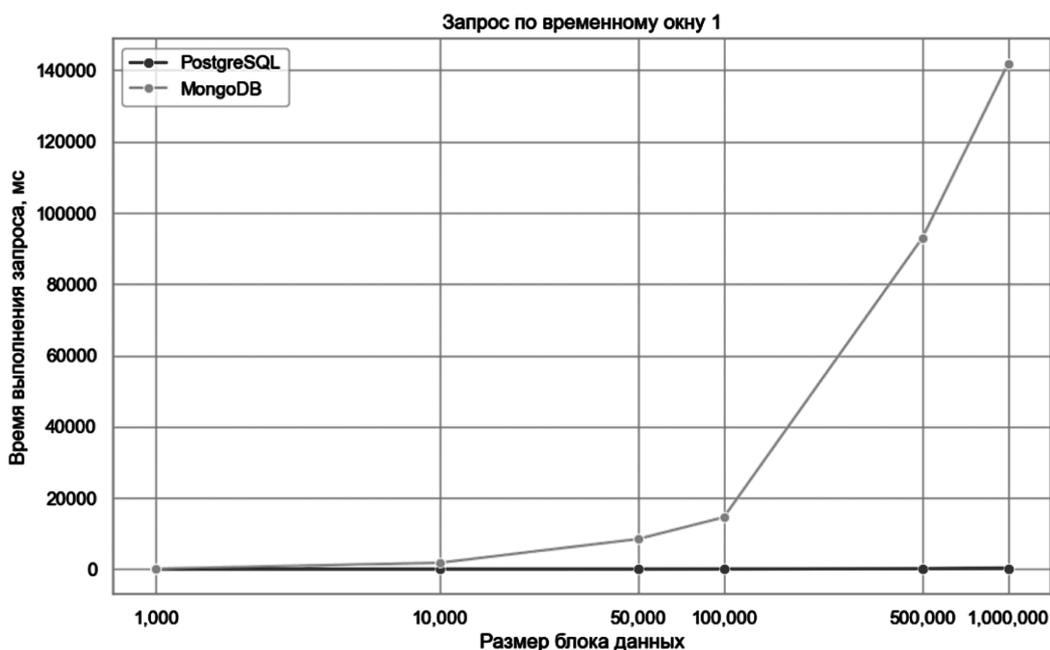


Рис. 5. Зависимость времени выполнения запроса по временному окну 1 от размера блока данных и СУБД

Таблица 3.

Зависимость времени выполнения запроса по временному окну 1 от размера блока данных и СУБД

| N записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 46                                | 121                            |
| 10000             | 40                                | 1791                           |
| 50000             | 52                                | 8477                           |
| 100000            | 76                                | 14555                          |
| 500000            | 163                               | 92965                          |
| 1000000           | 269                               | 141723                         |

Таблица 4.

Зависимость времени выполнения запроса по временному окну 2 от размера блока данных и СУБД

| N записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 35                                | 109                            |
| 10000             | 46                                | 1044                           |
| 50000             | 71                                | 5057                           |
| 100000            | 116                               | 9742                           |
| 500000            | 610                               | 50717                          |
| 1000000           | 1095                              | 99761                          |

Таблица 5.

Зависимость времени выполнения запроса агрегации 1 от размера блока данных и СУБД

| N записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 34                                | 65                             |
| 10000             | 41                                | 550                            |
| 50000             | 71                                | 2645                           |
| 100000            | 93                                | 5267                           |
| 500000            | 252                               | 26623                          |
| 1000000           | 857                               | 49854                          |

Таблица 6.

Зависимость времени выполнения запроса агрегации 2 от размера блока данных и СУБД

| N записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 34                                | 24                             |
| 10000             | 40                                | 46                             |
| 50000             | 65                                | 76                             |
| 100000            | 108                               | 147                            |
| 500000            | 396                               | 1643                           |
| 1000000           | 857                               | 5544                           |

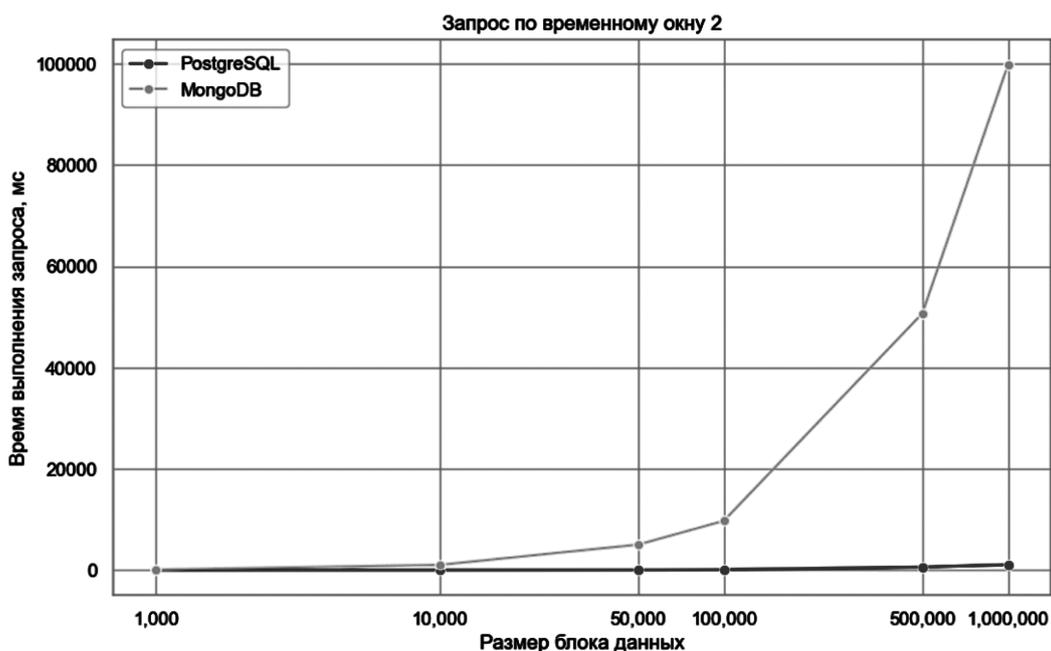


Рис. 6. Зависимость времени выполнения запроса по временному окну 2 от размера блока данных и СУБД

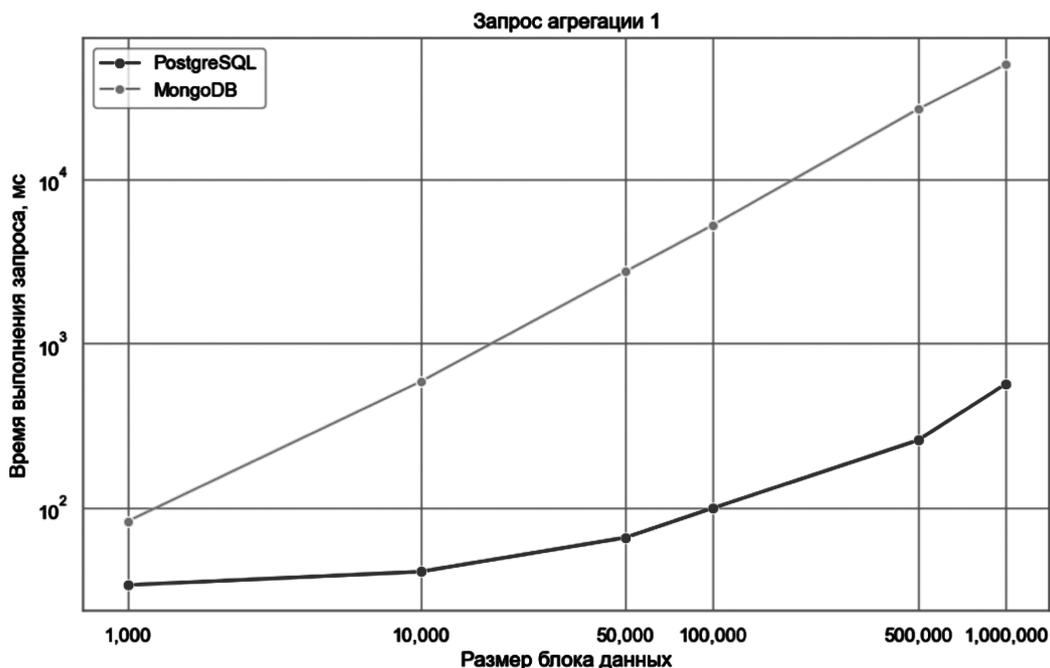


Рис. 7. Зависимость времени выполнения запроса агрегации 1 от размера блока данных и СУБД

- «Получить количество событий по типу за последние 10 часов с группировкой по пользователю и устройству» (Запрос по временному окну 2, рис. 6, табл. 4)

PostgreSQL продемонстрировала лучшие результаты благодаря эффективной обработке временных меток с использованием индексов и оптимизации выполнения группировок. MongoDB на крупных объемах данных показала значительно более высокое время выполнения

из-за затрат на обработку временных диапазонов в документной модели.

Для вычисления значений по параметру «время выполнения агрегационных запросов на блоке данных», были сформированы два запроса:

- «Получить всех пользователей, с событием покупки» (Запрос агрегации 1, рис. 7, табл. 5)
- «Найти количество уникальных пользователей для каждого типа события». (Запрос агрегации 2, рис. 8, табл. 6)

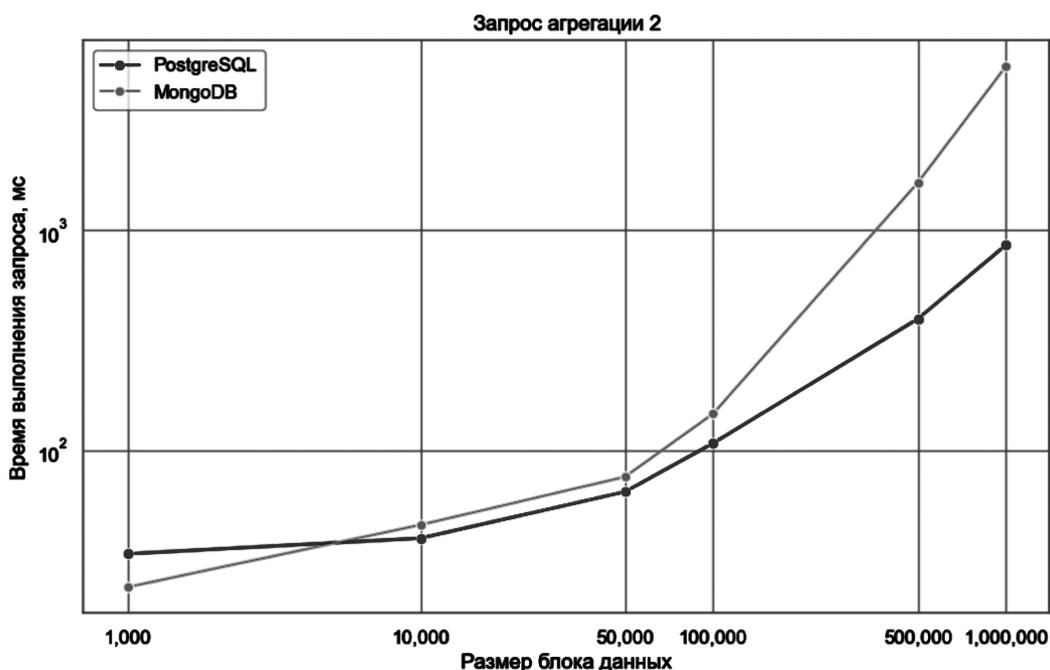


Рис. 8. Зависимость времени выполнения запроса агрегации 2 от размера блока данных и СУБД

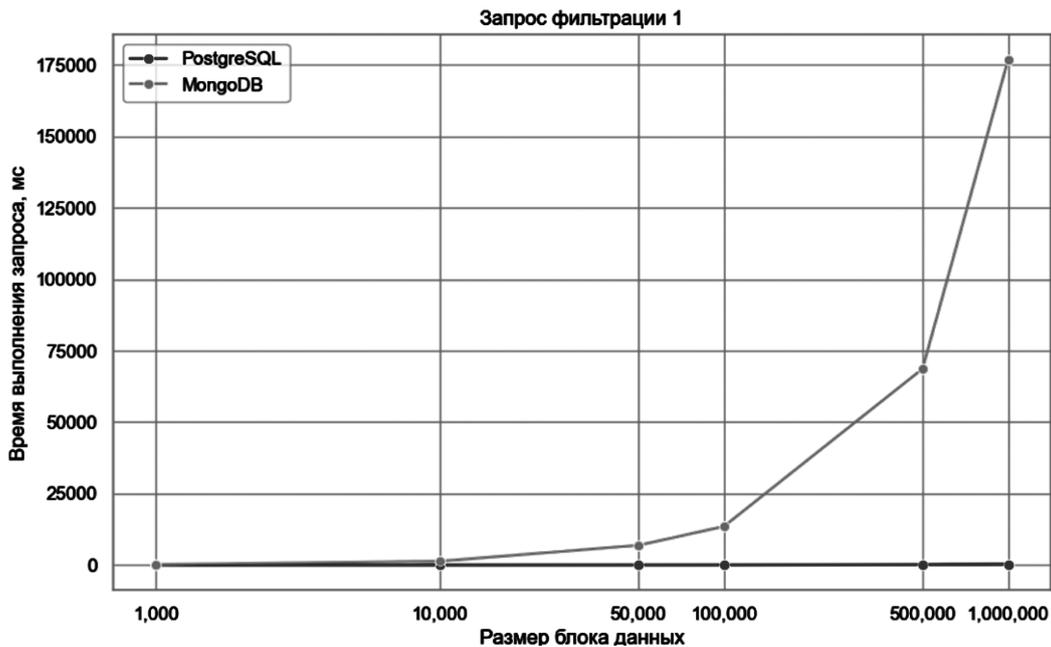


Рис. 9. Зависимость времени выполнения запроса фильтрации 1 от размера блока данных и СУБД

Таблица 7.

Зависимость времени выполнения запроса фильтрации 1 от размера блока данных и СУБД

| № записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 37                                | 185                            |
| 10000             | 39                                | 1340                           |
| 50000             | 55                                | 6903                           |
| 100000            | 76                                | 13519                          |
| 500000            | 188                               | 68653                          |
| 1000000           | 303                               | 176723                         |

При выполнении данных запросов агрегации значительное превосходство показывает PostgreSQL. В первом рассмотренном запросе PostgreSQL показывает более умеренный рост времени выполнения операции, а также выполняет ее за численно меньшее количество времени. Во втором запросе, несмотря на лучшие показатели у MongoDB на блоке из 1000 записей, на всех других блоках PostgreSQL показывает лучшие результаты. PostgreSQL показала значительное превосходство благодаря высокой эффективности механизмов группировки и агрегации. MongoDB, несмотря на гибкость своей модели, оказалась менее оптимизированной для таких операций, особенно на больших объемах данных.

Для вычисления значений по параметру «время выполнения фильтрующих запросов на блоке данных», были сформированы два запроса:

- «Найти пользователей, которые использовали

Таблица 8.

Зависимость времени выполнения запроса фильтрации 2 от размера блока данных и СУБД

| № записей в блоке | Время выполнения (PostgreSQL), мс | Время выполнения (MongoDB), мс |
|-------------------|-----------------------------------|--------------------------------|
| 1000              | 57                                | 164                            |
| 10000             | 58                                | 1695                           |
| 50000             | 52                                | 7526                           |
| 100000            | 69                                | 12329                          |
| 500000            | 214                               | 70444                          |
| 1000000           | 385                               | 127026                         |

устройства с операционной системой «Windows» (Запрос фильтрации 1, рис. 9, табл. 7)

- Найти все события типа «login» для устройств с операционной системой «Linux» (Запрос фильтрации 1, рис. 10, табл. 8)

В запросах фильтрации PostgreSQL также превосходит MongoDB. Несмотря на то, что разница на блоках 1000 и 1000 незначительна, на остальных блоках MongoDB сильно уступает в скорости. Это объясняется эффективностью индексных структур реляционных баз данных, которые минимизируют затраты на обработку сложных условий.

#### Оценка полученных результатов

Произведем анализ полученных данных по диапазонам (табл. 9, 10, рис. 11).

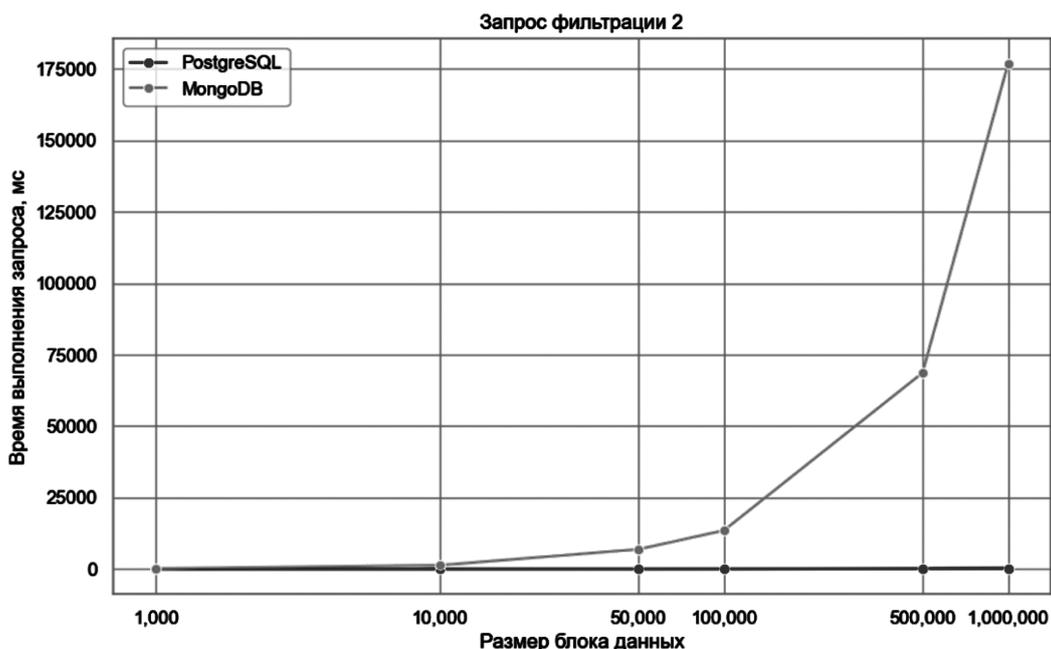


Рис. 10. Зависимость времени выполнения запроса фильтрации 2 от размера блока данных и СУБД

Таблица 9.

Зависимость времени выполнения запросов от размера блока данных для PostgreSQL

| N       | $t_w$ , мс | $t_r$ , мс | $t_{f1}$ , мс | $t_{f2}$ , мс | $t_{a1}$ , мс | $t_{a2}$ , мс | $t_{f1}$ , мс | $t_{f2}$ , мс |
|---------|------------|------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 1000    | 74         | 35         | 46            | 35            | 34            | 34            | 37            | 57            |
| 10000   | 694        | 51         | 40            | 46            | 41            | 40            | 39            | 58            |
| 50000   | 4016       | 117        | 52            | 71            | 71            | 65            | 55            | 52            |
| 100000  | 9737       | 198        | 76            | 116           | 93            | 108           | 76            | 69            |
| 500000  | 47633      | 898        | 163           | 610           | 252           | 396           | 188           | 214           |
| 1000000 | 102065     | 1931       | 269           | 1095          | 430           | 857           | 303           | 385           |

Таблица 10.

Зависимость времени выполнения запросов от размера блока данных для MongoDB

| N       | $t_w$ , мс | $t_r$ , мс | $t_{f1}$ , мс | $t_{f2}$ , мс | $t_{a1}$ , мс | $t_{a2}$ , мс | $t_{f1}$ , мс | $t_{f2}$ , мс |
|---------|------------|------------|---------------|---------------|---------------|---------------|---------------|---------------|
| 1000    | 76         | 2          | 121           | 109           | 65            | 24            | 185           | 164           |
| 10000   | 232        | 2          | 1791          | 1044          | 550           | 46            | 1340          | 1695          |
| 50000   | 1073       | 3          | 8477          | 5057          | 2645          | 76            | 6903          | 7526          |
| 100000  | 2212       | 3          | 14555         | 9742          | 5267          | 147           | 13519         | 12329         |
| 500000  | 11821      | 3          | 92965         | 50717         | 26623         | 1643          | 68653         | 70444         |
| 1000000 | 22692      | 3          | 141723        | 99761         | 49854         | 5544          | 176723        | 127026        |

Легенда таблиц 9, 10:

- $t_w$ , мс — время выполнения записи
- $t_r$ , мс — время выполнения чтения
- $t_{f1}$ , мс — время выполнения запроса по временному окну 1
- $t_{f2}$ , мс — время выполнения запроса по временному окну 2
- $t_{a1}$ , мс — время выполнения запроса по агрегации 1
- $t_{a2}$ , мс — время выполнения запроса по агрегации 2
- $t_{f1}$ , мс — время выполнения запроса по фильтрации 1
- $t_{f2}$ , мс — время выполнения запроса по фильтрации 2

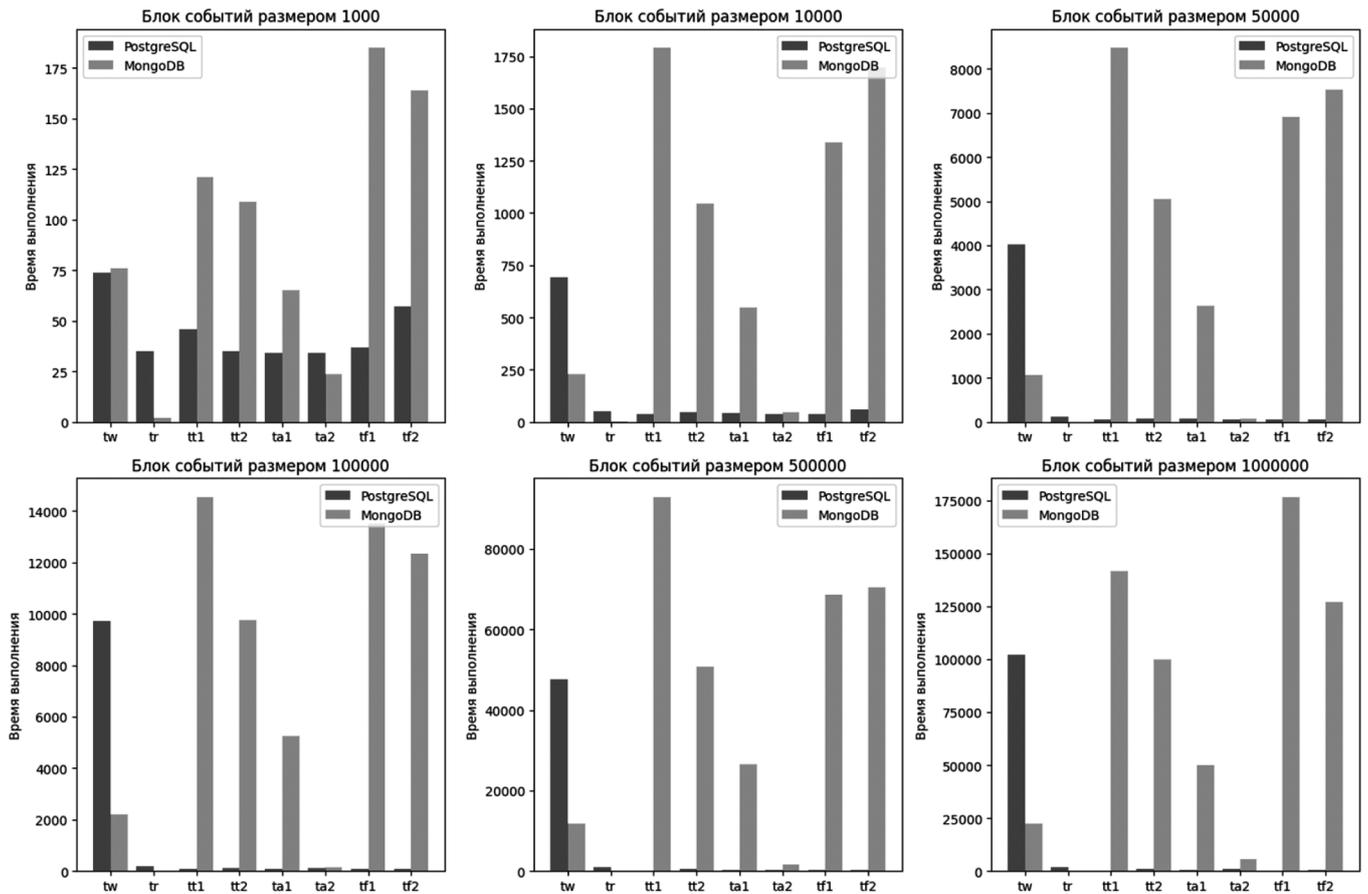


Рис. 11. Графики времени выполнения запросов для MongoDB и PostgreSQL

Из полученных результатов выполнения запросов следует превосходство MongoDB в запросах по записи и чтению. В случае записи, на всех значениях помимо 1000, PostgreSQL показывает гораздо более высокую численно скорость записи. В случае же чтения MongoDB показывает практически статическое время выполнения запроса для любого блока данных.

Однако, PostgreSQL показывает гораздо более высокие показатели при работе с запросами агрегации, фильтрации и запросам по временному окну. Данные операции не превышают 1.1 секунды даже для миллиона объектов, в то время как аналогичные запросы в MongoDB занимают от 5 до 176 секунд.

Таким образом, выбор СУБД и типа БД для хранения и обработки событийных данных напрямую зависит от характера работы с ними. При большом потоке данных и невысокой частоте аналитических запросов, оптимальной СУБД является MongoDB. При большом количестве аналитических забросов в короткий период времени, оптимальнее будет использовать PostgreSQL. Однако, важно отметить, что недостатки СУБД, выявлен-

ные в данном исследовании, возможно частично компенсировать механизмами индексирования и шардирования, предоставляемыми обеими СУБД [1] [2].

### Заключение

В данной работе было проведено исследование производительности СУБД при работе с событийными данными — документной MongoDB и реляционной PostgreSQL. Были выделены критерии и параметры, отвечающие за оценку производительности и на их основе проведено экспериментальное исследование. Оценка результатов исследования показала, что выбор СУБД зависит от реальных условий системы, для которой она послужит хранилищем данных: PostgreSQL, как представитель SQL БД, наилучшим образом покажет себя в системах, требующих в первую очередь быстрой обработки событийных данных и использующих аналитические запросы для построения мониторинга на их основе. MongoDB, как представитель NoSQL БД будет оптимальна для систем, в которых аналитические данные по событиям собираются нерегулярно, однако сами события в системе возникают в больших количествах.

---

ЛИТЕРАТУРА

1. Воропаев Д.В. Шардирование базы данных / Д.В. Воропаев, В.С. Кортун // Молодежь и наука: актуальные проблемы фундаментальных и прикладных исследований: Материалы VII Всероссийской национальной научной конференции молодых учёных, Комсомольск-на-Амуре, 08–12 апреля 2024 года. — Комсомольск-на-Амуре: Комсомольский-на-Амуре государственный университет, 2024. — С. 293–296. — EDN UFYNGH.
2. Abbasi Maryam & Bernardo Marco & Váz Paulo & Silva Jose & Martins Pedro. (2024). Revisiting Database Indexing for Parallel and Accelerated Computing: A Comprehensive Study and Novel Approaches. Information. 15. 429. 10.3390/info15080429. DOI: 10.3390/info15080429
3. Fischer Fabian & Fischer@uni, Fabian & Mansmann, Florian & Mansmann@uni Florian & Keim Daniel & Keim Daniel. (2011). Real-Time Visual Analytics for Event Data Streams. Proceedings of the ACM Symposium on Applied Computing. 10.1145/2245276.2245432. DOI: 10.1145/2245276.2245432
4. Gurcan Fatih; Berigel Muhammet. 2018. Gurcan Fatih; Berigel Muhammet. (2018). [IEEE 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) — Ankara, Turkey (2018.10.19-2018.10.21)] 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT) — Real-Time Processing of Big Data Streams: Lifecycle, Tools, Tasks, and Challenges., (), 1–6. DOI: 10.1109/ISMSIT.2018.8567061

---

© Горячкин Борис Сергеевич (bsgor@mail.ru); Панов Максим Константинович (panovmk@student.bmstu.ru)  
Журнал «Современная наука: актуальные проблемы теории и практики»