

ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: <https://doi.org/10.60797/IRJ.2025.161.48>

ВЛИЯНИЕ ХАРАКТЕРИСТИК ОПЕРАТИВНОЙ ПАМЯТИ DDR4 НА ПРОИЗВОДИТЕЛЬНОСТЬ СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ POSTGRESQL

Научная статья

Нуриев М.Г.^{1,*}, Хафизова А.Ш.², Пикулева Н.И.³, Гарифзянова Г.Г.⁴

¹ ORCID : 0009-0003-0741-1734;

⁴ ORCID : 0000-0003-2252-5812;

^{1, 2, 3} Казанский национальный исследовательский технический университет им. А. Н. Туполева – КАИ, Казань, Российская Федерация

⁴ Казанский национальный исследовательский технологический университет, Казань, Российская Федерация

* Корреспондирующий автор (mgnuriev[at]kai.ru)

Аннотация

В данной статье представлено комплексное исследование влияния аппаратных характеристик оперативной памяти DDR4 на производительность системы управления базами данных PostgreSQL. Авторы ставят целью определить степень влияния таких параметров оперативного запоминающего устройства как тактовая частота, тайминги включая задержки CAS, RAS-to-CAS, RP, RAS и пропускная способность на скорость выполнения SQL-запросов различной степени сложности. Для проведения экспериментов использовалась аппаратная конфигурация на базе процессора AMD Ryzen 5 3600 с модулями памяти DDR4, тестируемыми в двух режимах работы: XMP-3200 с частотой 3200 МГц, а также в режиме пониженной частоты 1333 МГц с оптимизированными таймингами. Методика тестирования включала проведение синтетических тестов с использованием утилиты WinRAR для измерения пропускной способности и задержек, нагрузочных тестов в PostgreSQL с применением инструмента pgbench для измерения количества транзакций в секунду, а также выполнение сложных SQL-запросов с операциями агрегации, соединениями таблиц через JOIN и генерацией данных с использованием Common Table Expressions и функции generate_series. Ключевые результаты исследования показали, что снижение частоты памяти с 3200 МГц до 1333 МГц привело к уменьшению производительности системы на 4,9 процента, выразившемуся в снижении количества транзакций в секунду с 12492 до 11876 и увеличении средней задержки с 1,596 миллисекунды до 1,679 миллисекунды. При этом было установлено, что тайминги памяти, выраженные в наносекундах, оказывают меньшее влияние на общую производительность по сравнению с частотными характеристиками, особенно при выполнении сложных запросов с высокими требованиями к пропускной способности. Вопрос экономической целесообразности перехода на высокочастотные модули DDR4 требует дифференцированного подхода и зависит от конкретной рабочей нагрузки: для серверов с интенсивными OLTP-запросами такие инвестиции могут быть полностью оправданы, тогда как для систем с умеренной нагрузкой разница в производительности может оказаться незначительной. Практическая значимость проведенного исследования заключается в возможности формулирования конкретных рекомендаций по выбору оперативной памяти для серверов PostgreSQL, а также в разработке методики тестирования производительности системы управления базами данных при различных конфигурациях памяти.

Ключевые слова: DDR4, PostgreSQL, оперативная память, производительность СУБД, тактовая частота, тайминги памяти, пропускная способность, pgbench, SQL-запросы, OLTP, аппаратная оптимизация, экономическая эффективность.

THE INFLUENCE OF DDR4 RAM CHARACTERISTICS ON THE PERFORMANCE OF THE POSTGRESQL DATABASE MANAGEMENT SYSTEM

Research article

Nuriev M.G.^{1,*}, Khafizova A.S.², Pikuleva N.I.³, Garifzianova G.G.⁴

¹ ORCID : 0009-0003-0741-1734;

⁴ ORCID : 0000-0003-2252-5812;

^{1, 2, 3} Kazan National Research Technical University named after A. N. Tupolev – KAI, Kazan, Russian Federation

⁴ Kazan National Research Technological University, Kazan, Russian Federation

* Corresponding author (mgnuriev[at]kai.ru)

Abstract

This article presents a complex study of the impact of DDR4 RAM hardware characteristics on the performance of the PostgreSQL database management system. The authors aim to determine the degree of influence of such RAM parameters as clock frequency, timings including CAS, RAS-to-CAS, RP, RAS delays, and bandwidth on the speed of executing SQL queries of varying degrees of complexity. The experiments were conducted using a hardware configuration based on an AMD Ryzen 5 3600 processor with DDR4 memory modules tested in two modes: XMP-3200 with a frequency of 3200 MHz, and a reduced frequency of 1333 MHz with optimised timings. The testing methodology included synthetic tests using the WinRAR utility to measure throughput and latency, load tests in PostgreSQL using the pgbench tool to measure the number of transactions per second, as well as performing complex SQL queries with aggregation operations, table joins via JOIN, and data generation using Common Table Expressions and the generate_series function. Key findings of the research showed that reducing the memory frequency from 3200 MHz to 1333 MHz resulted in a 4.9% decrease in system performance, resulting in a decrease in

transactions per second from 12,492 to 11,876 and an increase in average latency from 1.596 milliseconds to 1.679 milliseconds. At the same time, it was found that memory timings, expressed in nanoseconds, have less impact on overall performance than frequency characteristics, especially when executing complex queries with high bandwidth requirements. The question of the economic feasibility of switching to high-frequency DDR4 modules requires a differentiated approach and depends on the specific workload: for servers with intensive OLTP queries, such investments may be fully reasonable, while for systems with moderate loads, the difference in performance may be insignificant. The practical significance of the study lies in the ability to formulate specific recommendations for selecting RAM for PostgreSQL servers, as well as in the development of a methodology for testing the performance of database management systems with different memory configurations.

Keywords: DDR4, PostgreSQL, internal shortage, DBMS performance, clock speed, memory timings, bandwidth, pgbench, SQL queries, OLTP, hardware optimisation, cost-effectiveness.

Введение

С развитием информационных технологий всё более остро стоит проблема эффективной обработки больших объёмов данных. Современные системы управления базами данных (СУБД) всё чаще сталкиваются с необходимостью обеспечивать высокую скорость выполнения запросов при одновременном увеличении объёмов хранимой информации. Одним из ключевых факторов, влияющих на производительность таких систем, является аппаратная конфигурация вычислительной платформы, в особенности характеристики оперативной памяти.

Оперативная память типа Double Data Rate Synchronous Dynamic Random-Access Memory (DDR SDRAM [1], [2]) играет важную роль в обеспечении быстрого доступа к данным, используемым процессором. С развитием технологии появилось несколько поколений этой памяти, каждое из которых предлагает улучшенные параметры: тактовая частота, тайминги, пропускная способность. Однако вопрос о том, насколько значимо влияние этих параметров на реальную производительность СУБД при обработке запросов, остаётся открытым [3]. Более того, с экономической точки зрения не всегда очевидно, насколько целесообразно приобретение более дорогих модулей памяти с улучшенными характеристиками.

Целью данной работы является исследование влияния аппаратных характеристик оперативной памяти типа DDR4 на скорость выполнения SQL-запросов в системе управления базами данных PostgreSQL [4]. На основе экспериментальных данных предполагается оценить степень влияния таких параметров памяти, как частота, тайминги и пропускная способность, а также определить экономическую целесообразность перехода на более высокопроизводительные модули памяти.

Для достижения поставленной цели решаются следующие задачи:

1. Провести анализ устройства и принципов функционирования оперативной памяти DDR4, а также особенностей взаимодействия СУБД PostgreSQL с оперативной памятью.
2. Разработать методику тестирования производительности СУБД PostgreSQL при различных параметрах оперативной памяти.
3. Провести серию экспериментов по измерению времени выполнения SQL-запросов с использованием модулей памяти DDR4 с различными частотами, таймингами и уровнями пропускной способности.
4. Обработать и проанализировать полученные данные для выявления зависимости скорости выполнения запросов от характеристик ОЗУ.

5. Сделать выводы о степени влияния аппаратных параметров памяти на производительность СУБД и оценить экономическую целесообразность использования более высокочастотных модулей памяти.

Методика тестирования, представленная в исследовании, опирается на системный и всеобъемлющий подход к оценке влияния различных параметров оперативной памяти DDR4 на производительность системы управления базами данных PostgreSQL. Целью экспериментов является всестороннее изучение того, каким образом такие технические характеристики памяти, как тактовая частота, величины основных задержек (включая CAS, RAS-to-CAS, RP и RAS), а также пропускная способность, отражаются на скорости выполнения запросов, в частности — при сложных вычислительных сценариях.

В качестве базовой платформы был выбран современный вычислительный комплекс, включающий в себя центральный процессор AMD Ryzen 5 3600 с архитектурой Zen 2, способный обеспечивать высокую вычислительную мощность, совместно с четырьмя одноранговыми модулями DDR4-памяти объёмом по 8 гигабайт каждый. При этом была предусмотрена возможность работы оперативной памяти в двух режимах: в высокочастотной конфигурации XMP-3200 (с эффективной частотой 3200 мегагерц и таймингами 16-18-18-36 при напряжении 1,35 Вольта) и в режиме пониженной частоты 1333 мегагерца с ручной настройкой более агрессивных таймингов. Аппаратная платформа базировалась на материнской плате Asus Prime B450M-A II, а операционной системой выступала Microsoft Windows 10 версии 22H2. В качестве постоянного хранилища данных использовался твердотельный накопитель Samsung 980 EVO объёмом 500 гигабайт, заполненный на 70% для приближения условий к реальным эксплуатационным.

Тестирование включало как синтетические, так и прикладные этапы. На синтетическом уровне использовалась встроенная функция тестирования из утилиты WinRAR, которая посредством многопоточной обработки данных моделировала интенсивную нагрузку на память и процессор. В ходе этой процедуры измерялась скорость выполнения операций сжатия и распаковки данных, а также выявлялись возможные ошибки, указывающие на нестабильность работы памяти в той или иной конфигурации. Полученные результаты отражали производительность в килобайтах в секунду и демонстрировали общее влияние параметров памяти на обработку повседневных задач.

Для моделирования реальных условий работы была сгенерирована полноценная база данных объёмом 4 гигабайта с помощью библиотеки Faker. Структура базы данных была нормализована, снабжена внешними ключами и включала таблицы, предназначенные для выполнения сложных SQL-запросов, в которых активно использовались операции

JOIN, обобщённые табличные выражения (CTE), оконные функции и генерация последовательностей через функцию `generate_series`. В процессе тестирования применялись как встроенные средства мониторинга PostgreSQL, так и инструменты EXPLAIN и EXPLAIN ANALYZE, позволившие детально рассмотреть план выполнения запросов, оценить реальное время выполнения, количество обработанных строк и другие метрики.

Для моделирования высокой нагрузки был написан запрос, в котором использовались многочисленные агрегации, сортировки и фильтрации, предъявляющие высокие требования к оперативной памяти, особенно в части создания временных хэш-таблиц и подсчёта уникальных значений через COUNT(DISTINCT). Это позволило оценить пределы возможностей системы в условиях интенсивной обработки данных. Во всех тестах фиксировались ключевые показатели, включая количество транзакций в секунду и среднюю задержку обработки запросов. Тестирование проводилось многократно, при этом каждый запуск производился в идентичных условиях для исключения случайных флуктуаций производительности.

Все изменения параметров памяти производились вручную через UEFI BIOS, а стабильность системы дополнительно проверялась в стрессовых условиях. В ходе эксперимента отдельно фиксировались значения задержек в наносекундах для каждой конфигурации, что позволило рассчитать реальное влияние на скорость доступа. Особенно важно, что уменьшение частоты памяти приводило к существенному увеличению задержек и снижению пропускной способности, что напрямую отражалось на итоговой производительности как при синтетических тестах, так и при работе СУБД.

Методика, применённая в исследовании, представляет собой комплексный подход, сочетающий аппаратное тестирование, программное моделирование, анализ реальных SQL-нагрузок и количественную оценку всех изменений в производительности. Такой формат позволяет с высокой степенью достоверности оценить влияние тех или иных характеристик оперативной памяти на работу баз данных и сформулировать обоснованные рекомендации по выбору аппаратного обеспечения в зависимости от характера предполагаемой нагрузки.

Результаты проведённого исследования позволяют не только оценить текущую ситуацию с применением DDR4-памяти в серверах баз данных, но и получить представление о том, какие изменения в производительности можно ожидать при переходе на новые или обратно на более старые поколения памяти.

Таким образом, работа будет иметь как практическую ценность, связанную с оптимизацией существующих систем, так и теоретическую позволяя делать обоснованные прогнозы относительно будущих технологий памяти.

Технология DDR SDRAM

Появление нового типа ОЗУ было продиктовано ростом производительности процессоров и увеличением объёмов данных. Таким образом, SDRAM уже не могла удовлетворить растущие требования. В 2000 году сообщество инженеров (Joint Electron Device Engineering Council, JEDEC), специализирующихся в области электронных устройств утвердила стандарт DDR SDRAM. JEDEC — это ведущая независимая отраслевая ассоциация, занимающаяся разработкой и стандартизацией технологий в области микроэлектроники, включая оперативную память. В контексте разработки и внедрения DDR SDRAM (и последующих поколений памяти) роль JEDEC является критически важной [5], [6].

В JEDEC входят сотни компаний, включая производителей полупроводников, разработчиков технологий и поставщиков оборудования (например, Samsung, Micron Technology, Intel, Advanced Micro Devices, NVIDIA, SK Hynix, Texas Instruments, Qualcomm).

Главной отличительной особенностью нового стандарта является одноимённый принцип DDR (Double Data Rate), реализация которого основана на считывании данных не только по фронту, но и по срезу тактового сигнала. Таким образом, если тактовая частота DDR SDRAM составляет 100 МГц, эффективная частота передачи данных будет 200 МГц (рисунок 1).

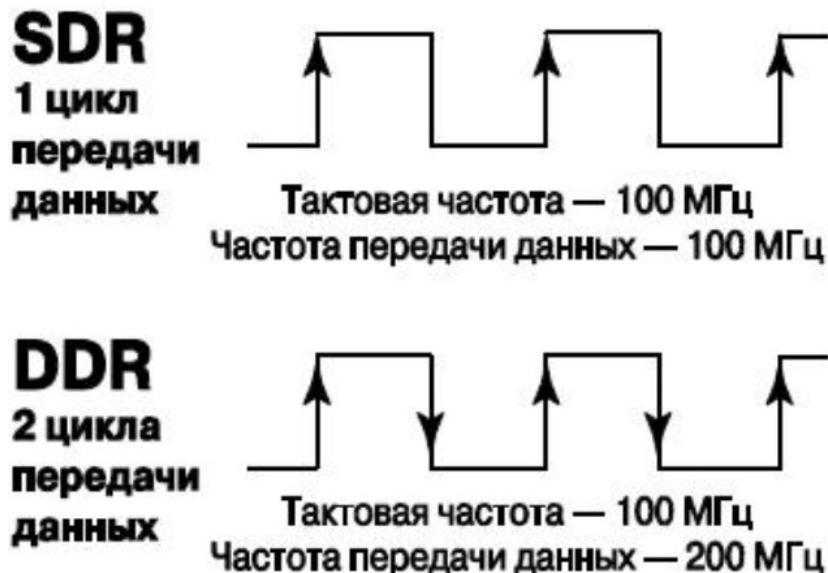


Рисунок 1 - Принцип работы DDR
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.1>

Частота, тайминги и напряжение — это основные характеристики модулей памяти, которые влияют на итоговую производительность электронных вычислительных машин [7], [8]. Рассмотрим подробнее тайминги.

Работа с оперативной памятью так или иначе сводится к физическому взаимодействию с электротехническими элементами. Все подобные процессы не могут происходить моментально и тайминги, непосредственно, регламентируют эти процессы позволяя всем компонентам памяти работать стабильно и предсказуемо. Единица измерения — такт. Чтобы считать или записать какие-либо значения в память, используются определённые сигналы:

- сигнал выбора столбца матрицы ячеек памяти (англ. Column Address Strobe, CAS);
- сигнал выбора строки матрицы ячеек памяти (англ. Row Address Strobe, RAS).

Рассмотрим 4 основных тайминга, которые оказывают наиболее ощутимое влияние на производительность модулей памяти.

Задержка строба адреса столбца (англ. CAS Delay, tCL) — задержка между выбором столбца и началом передачи данных. После отправки контроллером памяти запроса на доступ к определённой строке, он ожидает указанное число тактов, прежде чем данные будут доступны.

Задержка между активацией строки и выбором столбца (англ. RAS to CAS Delay, tRCD) — это время, необходимое для активации строки перед обращением ко столбцу. Перед тем как начать чтение или запись данных, контроллер памяти должен активировать строку (RAS) и затем выбрать столбец (CAS). Именно задержку между двумя этими операциями регламентирует tRCD.

Задержка между командой на подзарядку precharge до момента закрытия строки (англ. Row Precharge Time, tRP) — это время, необходимое для закрытия текущей строки перед открытием новой. После завершения операций с текущей строкой память должна быть «подготовлена» (precharged) для работы с другой строкой. Значение tRP определяет, сколько тактовых циклов требуется для этого процесса.

tRAS (Row Active Time) это время между активацией строки до срабатывания команды precharge.

Для простоты восприятия можно воспользоваться картой с рассчитанными значениями зависимостей времени доступа к памяти в наносекундах от частоты работы и задержек CAS (рисунок 2).

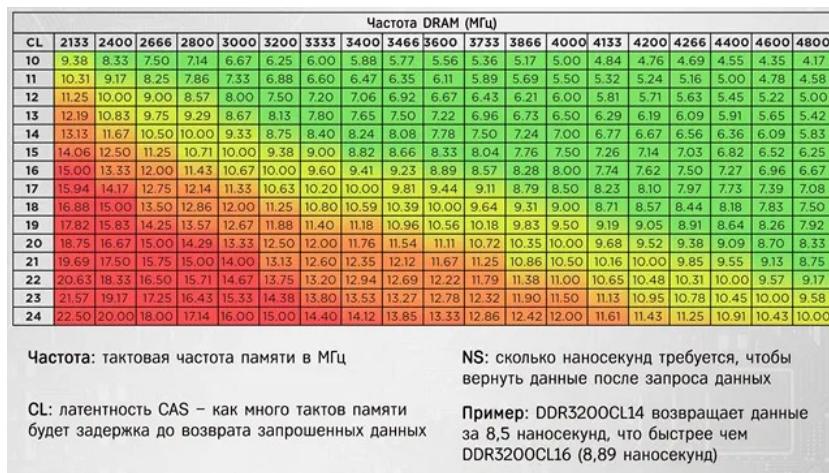


Рисунок 2 - Карта времени доступа к оперативной памяти

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.2>

Особенности работы PostgreSQL с оперативной памятью

PostgreSQL является дисковой СУБД, но она активно использует оперативную память для ускорения работы, что будет наглядно продемонстрировано.

PostgreSQL использует оперативную память для кэширования данных, обработки запросов, хранения промежуточных результатов операций.

В отличие от in-memory, СУБД PostgreSQL не хранит все данные в RAM по умолчанию, эти ограничения прописаны в конфигурационном файле, однако существуют множество механизмов для кэширования и оптимизации администрирования баз данных. Основными механизмами, которые взаимодействуют с оперативной памятью являются `shared_buffers`, `work_mem` и `maintenance_work_mem`.

- `shared_buffers` — параметр, определяющий, сколько оперативной памяти PostgreSQL выделяет для кэширования данных. Этот кэш используется для хранения страниц данных (блоков), которые читаются с диска или записываются на него;

- `work_mem` — параметр определяет сколько памяти может быть выделено для выполнения операций сортировки, хэширования и агрегации внутри одного запроса;

- `maintenance_work_mem` — параметр, который определяет, сколько памяти выделяется для операций обслуживания `ALTER TABLE`, `VACUUM` и создания индексов.

Существуют некоторые рекомендованные значения этих параметров, однако в целях повышения воспроизводимости и надежности эксперимента в работе используются свои настройки СУБД.

Ниже представлена диаграмма, визуализирующая распределение различных параметров по их удельному весу в общей совокупности. Каждый сектор данной диаграммы соответствует отдельной категории или параметру, при этом размер каждого сектора пропорционален относительной значимости или доле этого параметра в общем объеме анализируемых данных (рисунок 3).

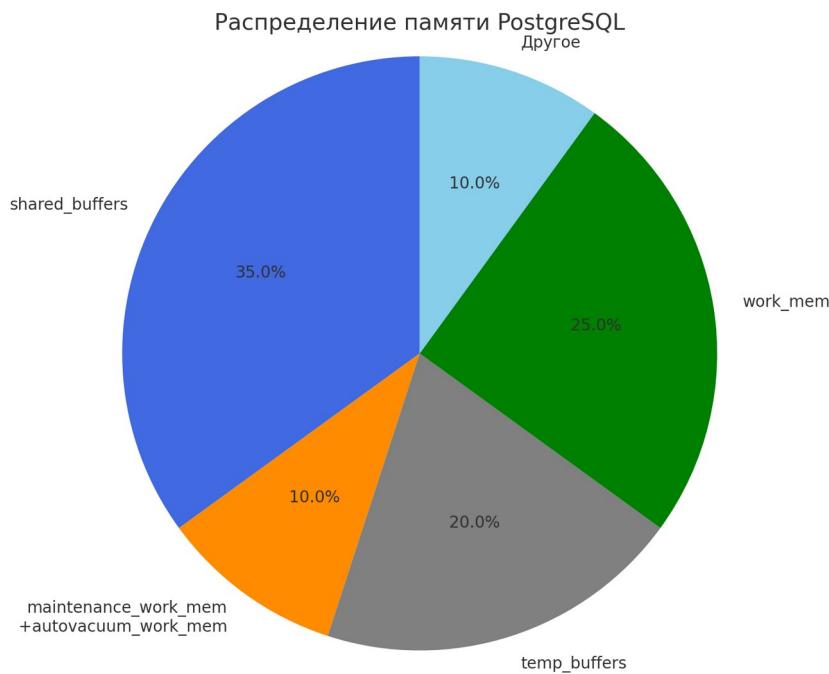


Рисунок 3 - Диаграмма относительных величин главных параметров памяти

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.3>

Выбор правильной конфигурации памяти является важным фактором для достижения максимально возможной производительности системы при сборке или модернизации сервера. Однако более подробный анализ экономической целесообразности использования высокопроизводительных модулей памяти следует проводить при тщательной оценке специфики и объёма выполняемых задач. Только при грамотном балансе между стоимостью, производительностью и реальными потребностями сервера можно добиться оптимального соотношения цены и качества, а также обеспечить долгосрочную эффективность и надёжность вычислительной системы.

Конфигурация выбранного аппаратного обеспечения

Аппаратно-программная конфигурация (рисунок 4) представлена рядовыми компонентами для современных персональных компьютеров (см. таблицу 1).

Processor	<table border="1"> <tr><td>Name</td><td colspan="3">AMD Ryzen 5 3600</td></tr> <tr><td>Code Name</td><td colspan="3">Matisse</td></tr> <tr><td>Package</td><td colspan="3">Max TDP 65.0 W</td></tr> <tr><td>Technology</td><td colspan="3">Socket AM4 (1331)</td></tr> <tr><td>Family</td><td>7 nm</td><td>Core Voltage</td><td>1.341 V</td></tr> <tr><td>Ext. Family</td><td>F</td><td>Model</td><td>1</td></tr> <tr><td>Instructions</td><td>17</td><td>Stepping</td><td>0</td></tr> <tr><td>Cache</td><td>Ext. Model</td><td>71</td><td>Revision</td></tr> <tr><td colspan="4">AMD Ryzen 5 3600 6-Core Processor</td></tr> <tr><td colspan="4">MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AMD-V, AES, AVX, AVX2, FMA3, SHA</td></tr> </table>				Name	AMD Ryzen 5 3600			Code Name	Matisse			Package	Max TDP 65.0 W			Technology	Socket AM4 (1331)			Family	7 nm	Core Voltage	1.341 V	Ext. Family	F	Model	1	Instructions	17	Stepping	0	Cache	Ext. Model	71	Revision	AMD Ryzen 5 3600 6-Core Processor				MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AMD-V, AES, AVX, AVX2, FMA3, SHA			
Name	AMD Ryzen 5 3600																																											
Code Name	Matisse																																											
Package	Max TDP 65.0 W																																											
Technology	Socket AM4 (1331)																																											
Family	7 nm	Core Voltage	1.341 V																																									
Ext. Family	F	Model	1																																									
Instructions	17	Stepping	0																																									
Cache	Ext. Model	71	Revision																																									
AMD Ryzen 5 3600 6-Core Processor																																												
MMX(+), SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, SSE4A, x86-64, AMD-V, AES, AVX, AVX2, FMA3, SHA																																												
Clocks (Core #0)	<table border="1"> <tr><td>Core Speed</td><td colspan="3">3949.08 MHz</td></tr> <tr><td>Multiplier</td><td colspan="3">x 39.5 (5.5 - 42.0)</td></tr> <tr><td>Bus Speed</td><td colspan="3">99.98 MHz</td></tr> <tr><td>Rated FSB</td><td colspan="3"></td></tr> </table>			Core Speed	3949.08 MHz			Multiplier	x 39.5 (5.5 - 42.0)			Bus Speed	99.98 MHz			Rated FSB																												
Core Speed	3949.08 MHz																																											
Multiplier	x 39.5 (5.5 - 42.0)																																											
Bus Speed	99.98 MHz																																											
Rated FSB																																												
Cache	<table border="1"> <tr><td>L1 Data</td><td>6 x 32 KBytes</td><td>8-way</td></tr> <tr><td>L1 Inst.</td><td>6 x 32 KBytes</td><td>8-way</td></tr> <tr><td>Level 2</td><td>6 x 512 KBytes</td><td>8-way</td></tr> <tr><td>Level 3</td><td>2 x 16 MBytes</td><td>16-way</td></tr> </table>			L1 Data	6 x 32 KBytes	8-way	L1 Inst.	6 x 32 KBytes	8-way	Level 2	6 x 512 KBytes	8-way	Level 3	2 x 16 MBytes	16-way																													
L1 Data	6 x 32 KBytes	8-way																																										
L1 Inst.	6 x 32 KBytes	8-way																																										
Level 2	6 x 512 KBytes	8-way																																										
Level 3	2 x 16 MBytes	16-way																																										
Selection	Socket #1	Cores	6																																									
		Threads	12																																									

Рисунок 4 - Спецификации центрального процессора

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.4>

Центральный процессор Ryzen 5 3600 построен на CISC архитектуре x86 Zen 2 и поддерживает работу только с модулями памяти типа DDR4 [9], [10] максимальной частотой 3200 МГц (рисунок 5).

Все исследования и тестирования проводятся при заводских значениях напряжений и тактовых частот самого процессора, изменения не предусмотрены. Это позволяет оценить реальную производительность в стандартных условиях эксплуатации.

Таблица 1 - Аппаратно-программная конфигурация

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.5>

Центральный процессор	AMD Ryzen 5 3600 4200 МГц
Графическая подсистема	AMD RX 6600 8 ГБ GDDR6
Подсистема памяти ОЗУ	DDR4 SDRAM 4x8 ГБ. Максимальная паспортная частота 3200 МГц
Системная плата	Asus Prime B450M-A II
Операционная система	Microsoft Windows 10 22H2
Твердотельный накопитель SSD M.2	Samsung 980 EVO 500ГБ

Memory Slot Selection					
Slot #1	DDR4	Module Size	8 GBytes		
	Max Bandwidth	SPD Ext.	XMP 2.0		
	Module Manuf.	Week/Year	01 / 24		
	DRAM Manuf.	Ranks	Single		
	Part Number	Correction			
	Serial Number	Registered			

Timings Table				
	JEDEC #8	JEDEC #9	XMP-3200	XMP-3002
Frequency	1200 MHz	1200 MHz	1600 MHz	1501 MHz
CAS# Latency	17.0	18.0	16.0	15.0
RAS# to CAS#	17	17	18	17
RAS# Precharge	17	17	18	17
tRAS	39	39	36	36
tRC	55	55	74	69
Command Rate				
Voltage	1.20 V	1.20 V	1.350 V	1.350 V

Рисунок 5 - Спецификации центрального процессора

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.6>

Все модули памяти представлены в одноранговом исполнении. Ранг – это логическая группа чипов памяти на модуле ОЗУ, которая подключена к одному каналу и управляется отдельным набором сигналов.

Одноранговый модуль содержит один логический набор чипов, который обрабатывает данные за один такт. Двухранговая же реализация подразумевает два независимых набора чипов, которые работают поочерёдно, но используют общую шину.

С точки зрения производительности одноранговые модули памяти несколько быстрее в некоторых сценариях, где важна высокая скорость доступа. Двухранговые же модули потенциально выгоднее использовать для серверов из-за более высоких пропускных способностей.

В дальнейших исследованиях используются именно одноранговые модули по причине их дешевизны и распространённости (рисунок 6).

Стоит отметить, что все модули памяти характеризуются одинаковым объемом, составляющим 8 ГБ. Данное решение обусловлено тем, что применение модулей различного объема может стать причиной снижения производительности системы при заполнении модуля с меньшим объемом возникает переход в одноканальный режим работы, что негативно сказывается на скорости передачи данных и общем быстродействии системы.

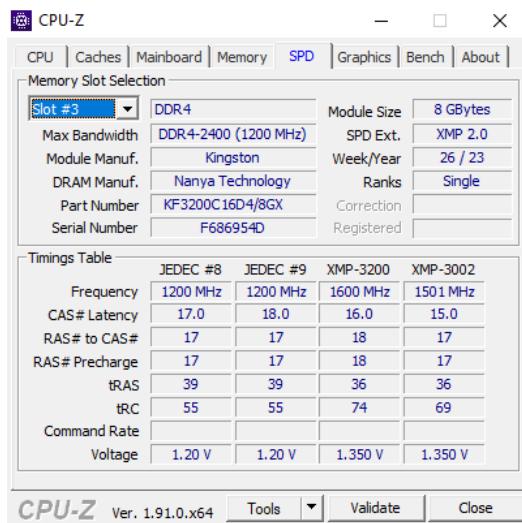


Рисунок 6 - Спецификации второй пары модулей памяти на чипах Nanya
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.7>

Даже при условии, что две пары модулей памяти изготовлены с использованием чипов различных производителей — таких как Micron Technology и Nanya Technology — их основные эксплуатационные характеристики [11], [12], включая тактовую частоту и тайминги, строго регламентированы и идентичны для каждого отдельного модуля. Это означает, что с точки зрения технических спецификаций, указанных в профиле, все модули соответствуют одному стандарту, что позволяет им потенциально работать совместно без значительных проблем.

Тем не менее стоит учитывать, что использование модулей памяти с различной элементной базой, особенно при задействовании всех четырёх слотов материнской платы в двухканальном режиме, может повлиять на стабильность работы всей системы. Дело в том, что несмотря на одинаковые номинальные параметры, физические различия между чипами памяти, такими как разница в технологических процессах изготовления, схемотехнике или даже реакции на изменение напряжения и температуры, могут привести к нестабильной работе модулей в составе одной конфигурации.

Значения пропускных способностей и задержек памяти в профиле XMP-3200 представлены на рисунке 7.

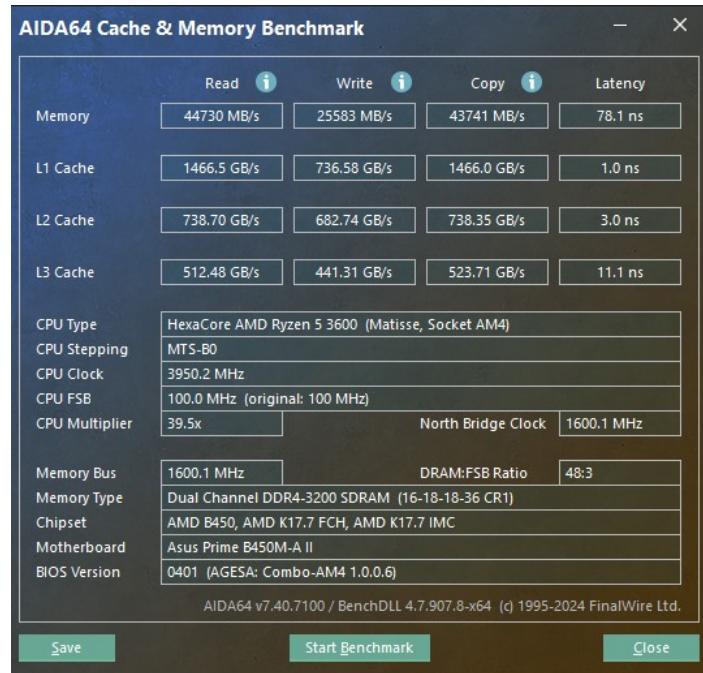


Рисунок 7 - Значения пропускных способностей и задержек памяти в профиле XMP-3200
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.8>

Номинальные параметры выставлены в соответствии с экстремальным профилем памяти (англ. Extreme Memory Profile, XMP) [13], [14], частота составляет 3200 МГц, тайминги (16, 18, 18, 36) при напряжении 1,35В.

Проведем первые измерения посредством выполнения сжатия и распаковки данных с помощью утилиты WinRAR.

Команда «Тест быстродействия» в утилите WinRAR позволяет сравнивать производительность алгоритма сжатия RAR на разных компьютерах [15], [16].

Она генерирует случайные данные, вызывающие повышенную нагрузку на процессор и память. Затем данные сжимаются и распаковываются по алгоритму RAR, после чего результат распаковки сравнивается с исходными данными. Если обнаруживается какое-либо различие, в окне команды в строке «Ошибки» выводится сообщение «Да». Такие ошибки могут свидетельствовать о проблемах с аппаратурой, например о нестабильной работе памяти. Кроме того, отображается объём обработанных данных и скорость сжатия — текущая и результирующая, в КБ/с. Результирующую скорость можно использовать для сравнения производительности RAR в различных условиях, например при выборе нового компьютера, чтобы узнать, какой из них быстрее сжимает данные. Используется общий алгоритм упаковки в режиме обычного сжатия со словарём 32 МБ, все дополнительные фильтры и алгоритмы отключаются.

С помощью параметра «Многопоточность» можно сравнить производительность обычной однопоточной и многопоточной (оптимизированной для мультипроцессорных архитектур) версий алгоритма сжатия RAR.

Для заполнения словаря упаковки, который в начале операции пуст, требуется некоторое время. Пока он не будет заполнен, значение скорости непостоянно, поэтому текущая скорость начинает отображаться лишь через несколько секунд после вызова команды.

Результирующая скорость выводится только после сбора статистики, необходимой для получения точного результата. После того как результирующая скорость установлена, она больше не изменяется.

Несмотря на то, что исходные данные случайны, степень их избыточности и другие параметры всегда останутся одинаковыми, поэтому команда будет выдавать практически постоянную текущую скорость вне зависимости от длительности выполнения при условии, что загрузка системы не изменяется.

Далее была изменена частотная характеристика и тайминги модулей памяти (рисунок 8). Все изменения производились через унифицированный расширяемый интерфейс встроенного программного обеспечения (англ. Unified Extensible Firmware Interface, UEFI [17], [18]). Следует отметить, что подобного рода изменения поддерживают не все материнские платы, а лишь конкретные чипсеты. Эту информацию требуется уточнять непосредственно в технических спецификациях платы.

При превышении паспортных значений частоты и таймингов модулей памяти, стабильная работа не гарантируется и требует дополнительных тестов стабильности под нагрузкой.

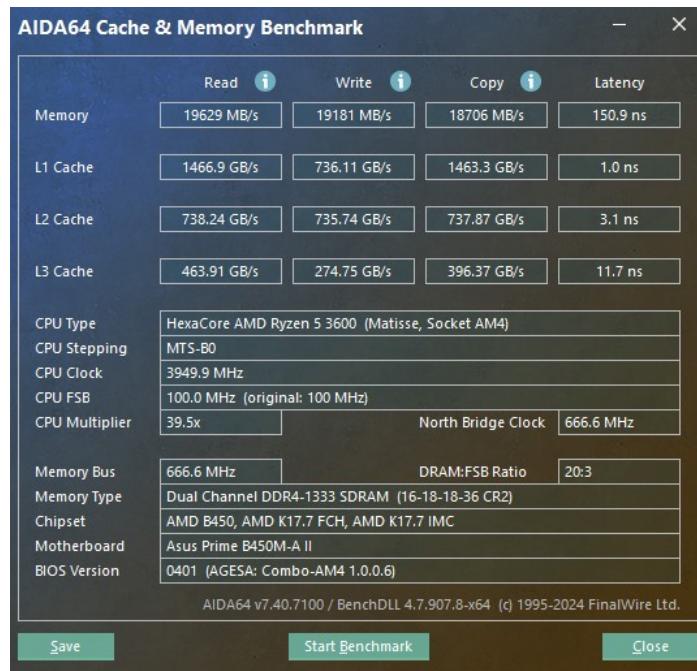


Рисунок 8 - Значения пропускных способностей и задержек памяти в конфигурации с частотой 1333 МГц, сниженными таймингами и напряжением
 DOI: <https://doi.org/10.60797/IRJ.2025.161.48.9>

После внесения изменений в конфигурацию модулей, отчётливо прослеживается значительное снижение пропускной способности до 19 ГБ/с и увеличение задержек при обращении к памяти вплоть до 150 наносекунд, что приводит к снижению производительности системы (рисунок 9).

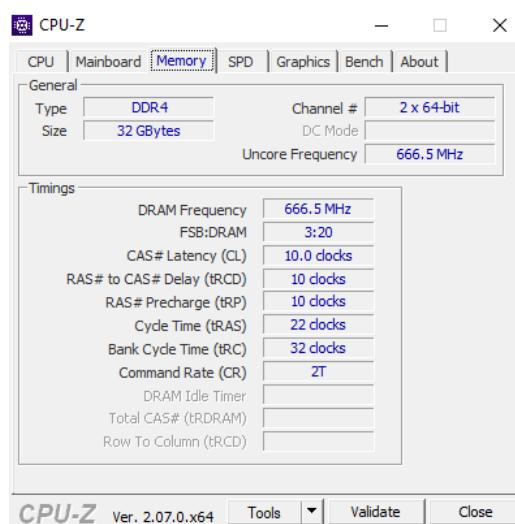


Рисунок 9 - Изменения таймингов памяти
 DOI: <https://doi.org/10.60797/IRJ.2025.161.48.10>

В результате снижения пропускной способности и роста задержек при работе с памятью возникают узкие места, которые замедляют выполнение операций и могут увеличить общее время отклика системы.

Снижение таймингов в нашем случае не приведет к увеличению скорости доступа из-за некоторых особенностей. Хотя тайминги в конфигурации XMP-3200 выше в тактах, но в пересчёте в наносекунды они всё равно оказываются ниже из-за более высокой эффективной частоты памяти.

Для расчёта используем формулу $T = \frac{t}{f} \times 1000$.

Тайминги в наносекундах для 3200 МГц:

- CL = 5 нс.

- tRCD = 5,625 нс.

- tRP = 5,625 нс.

- tRAS = 11,25 нс.

Тайминги в наносекундах для 1333 МГц:

- CL = 7,5 нс.
- tRCD = 7,5 нс.
- tRP = 7,5 нс.
- tRAS = 16,5 нс.

Уменьшение частоты памяти отразилось на производительности в тесте. Таким образом, результаты отличаются примерно на 14,43% в однопоточном режиме и на 20,06% (рисунок 10) в пользу конфигурации XMP-3200.

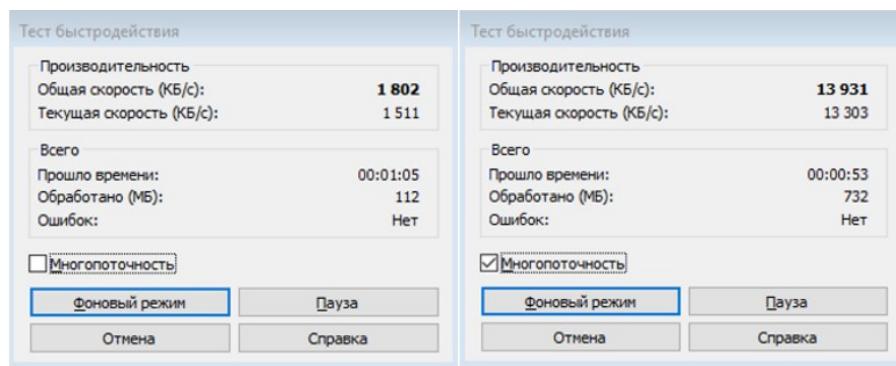


Рисунок 10 - Проведённый тест в многопоточном и однопоточном режиме в конфигурации 1333МГц

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.11>

Тест сжатия данных является простым, но информативным инструментом, в том числе из-за комплексной нагрузки на все компоненты вычислительной системы в повседневных задачах. Хотя данный тест не предоставляет исчерпывающих данных для оценки производительности, он хорошо зарекомендовал себя в качестве одного из элементов программного комплекса тестирования благодаря своей доступности и презентабельности.

Генерация базы данных и проведение измерений

Далее приступим к работе с системой управления базами данных PostgreSQL. Для этого нам понадобится установить саму СУБД с платформой pgAdmin для ее администрирования, а также связать с Python3 для дальнейшей генерации базы данных [19], [20].

Дальнейшая подготовка инфраструктуры требует обновления библиотек (рисунок 11).

```
C:\Users\Michael Suyagin\AppData\Local\Programs\Python\Python313>python.exe -m pip install --upgrade pip setuptools wheel
Requirement already satisfied: pip in c:\users\michael suyagin\appdata\local\programs\python\python313\lib\site-packages (24.2)
Collecting pip
  Using cached pip-24.3.1-py3-none-any.whl.metadata (3.7 kB)
Collecting setuptools
  Using cached setuptools-75.6.0-py3-none-any.whl.metadata (6.7 kB)
Collecting wheel
  Using cached wheel-0.45.1-py3-none-any.whl.metadata (2.3 kB)
Using cached pip-24.3.1-py3-none-any.whl (1.8 kB)
Using cached setuptools-75.6.0-py3-none-any.whl (1.2 kB)
Using cached wheel-0.45.1-py3-none-any.whl (72 kB)
Installing collected packages: wheel, setuptools, pip
  Attempting uninstall: pip
    Found existing installation: pip 24.2
    Uninstalling pip-24.2...
      Successfully uninstalled pip-24.2
    Successfully installed pip-24.3.1 setuptools-75.6.0 wheel-0.45.1
```

Рисунок 11 - Обновление зависимостей библиотек

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.12>

Обновление зависимостей библиотек необходимо для обеспечения совместимости с другими библиотеками, которые требуют pip и setuptools для правильной установки зависимостей. Конкретно в данном случае обновления потребовал только pip. Setuptools и wheel уже были в актуальных версиях.

Также было принято решение об использовании библиотеки asynpg для работы с СУБД.

Asynpg — это высокопроизводительная асинхронная библиотека для Python, предназначенная для работы с PostgreSQL. Она позволяет выполнять SQL-запросы к PostgreSQL без блокировки основного потока, что критично для асинхронных приложений (например, на базе asyncio, FastAPI или Quart) [21], [22].

PostgreSQL предоставляет гибкие возможности для настройки использования оперативной памяти. Правильная конфигурация параметров позволяет значительно повысить производительность системы, особенно при работе с большими объемами данных. Однако важно учитывать специфику рабочей нагрузки и характеристики сервера, чтобы избежать перегрузки системы и обеспечить стабильную работу. Значения параметров shared_buffers и work_mem по умолчанию крайне малы для проведения тестов такого рода и будут плохо влиять на результатирующую производительность в работе с большими данными и, следовательно, негативно отразятся на презентабельности всего исследования.

Таким образом, конфигурация по умолчанию неприменима для исследований скорости выполнения запросов из-за периодического обращения в постоянное запоминающее устройство. Значения соответствующих параметров можно изменить, отредактировав файл postgresql.conf. В данном случае было принято решение изменить значения на 4 ГБ для того, чтобы полностью покрыть потребности базы данных в адресном пространстве. Чтобы изменения были приняты,

нужно перезапустить службу postgresql-x64-17. Служба postgresql-x64-17 обеспечивает работу сервера PostgreSQL в операционной системе Windows. Запуск происходит в качестве сервиса ОС, что автоматизирует работу и предоставляет до-ступ для клиентских приложений.

PostgreSQL с pgAdmin предоставляет удобные и информативные метрики производительности (рисунок 12):

- Transactions (транзакции): синяя линия показывает общее количество транзакций, выполняемых в секунду.
- Commits (подтверждения): оранжевая линия отражает количество успешно завершенных транзакций.
- Rollbacks (откаты): зеленая линия показывает количество отмененных транзакций.
- Inserts (вставки): голубая линия показывает количество вставленных строк (кортежей).
- Updates (обновления): оранжевая линия отражает количество обновленных строк.
- Deletes (удаления): зеленая линия показывает количество удаленных строк.
- Fetched (полученные): голубая линия показывает количество строк, полученных в результате выполнения запросов.
- Returned (возвращенные): оранжевая линия отражает количество строк, возвращаемых клиентам.
- Reads (чтения блоков): голубая линия показывает количество блоков, считанных из диска.
- Hits (попадания в кэш): оранжевая линия отражает количество блоков, найденных в кэше без обращения к диску.

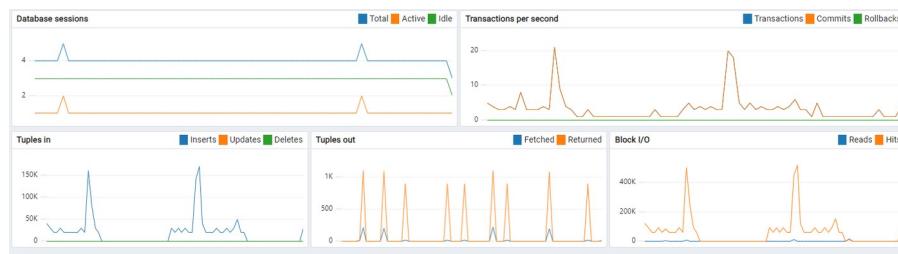


Рисунок 12 - Метрики производительности PostgreSQL

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.13>

Однако, из-за недостаточной точности и репрезентативности данного метода оценки производительности в непосредственных тестах было принято решение его не использовать.

Для генерации базы данных (рисунок 13) объёмом 4 ГБ написана программа на Python с применением библиотеки Faker, которая позволяет генерировать правдоподобные данные, для демонстрации наиболее репрезентативных результатов по итогам исследования.

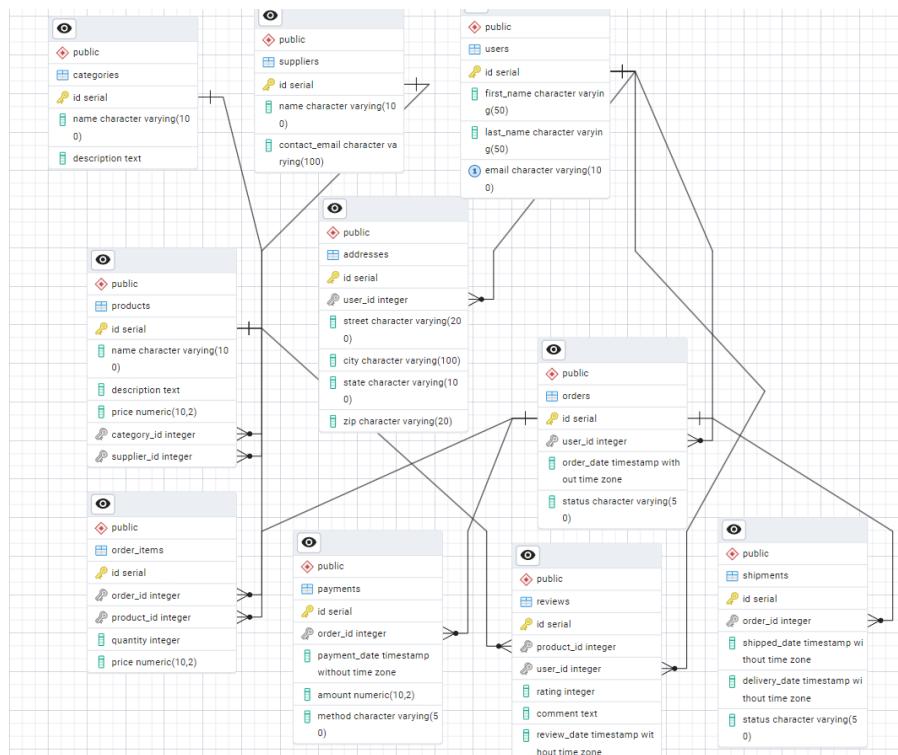


Рисунок 13 - Логическая модель базы данных

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.14>

База данных нормализована, что позволяет избежать дублирования данных и упрощает поддержку системы. Использование внешних ключей гарантирует целостность данных и позволяет легко связывать различные сущности, а результирующая структура позволяет эффективно выполнять сложные запросы.

Промежуточный вариант создания необходим для отладки процесса генерации базы данных, в связи с высокой нагрузкой на алгоритм Faker, который при достижении определённого количества записей перестаёт корректно функционировать. Обратим внимание, что на это ушло всего порядка 13 секунд.

Итоговый вариант базы данных объёмом 4 ГБ был записан на твердотельный накопитель Samsung SSD 980 EVO 500ГБ, чтобы обеспечить максимальную производительность, а также свести к минимуму недостатки дисковых СУБД.

Произведены измерения скорости последовательных и случайных чтений накопителя, где хранится база данных произведены в условиях его заполнения на 70%, что несколько снижает скорость доступа к памяти из-за заполнения Single Level Cell [23], [24] кэша (рисунок 14).

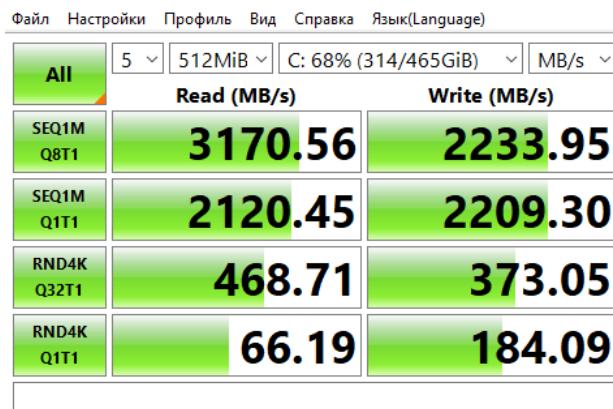


Рисунок 14 - Тестирование SSD, на котором находится база данных
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.15>

Далее проведём тестирование с помощью pgbench.

pgbench — это встроенный инструмент PostgreSQL для тестирования производительности базы данных. Он позволяет моделировать нагрузку на базу данных, выполнять транзакции и измерять такие метрики, как количество транзакций в секунду (англ. transactions per second, TPS). pgbench часто используется для оценки производительности системы (рисунок 15), тестирования настроек конфигурации или сравнения различных аппаратных конфигураций.

```
C:\Users\Michael Suyagin>pgbench -c 20 -j 4 -T 60 -P 10 -r -U postgres -d diploma
Password:
pgbench (17.2)
starting vacuum...end.
progress: 10.0 s, 11801.3 tps, lat 1.623 ms stddev 0.630, 0 failed
progress: 20.0 s, 12452.9 tps, lat 1.601 ms stddev 0.622, 0 failed
progress: 30.0 s, 12584.0 tps, lat 1.584 ms stddev 0.554, 0 failed
progress: 40.0 s, 12501.5 tps, lat 1.595 ms stddev 0.579, 0 failed
progress: 50.0 s, 12590.1 tps, lat 1.583 ms stddev 0.544, 0 failed
progress: 60.0 s, 12541.4 tps, lat 1.590 ms stddev 0.564, 0 failed
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 10
query mode: simple
number of clients: 20
number of threads: 4
maximum number of tries: 1
duration: 60 s
number of transactions actually processed: 744786
number of failed transactions: 0 (0.000%)
latency average = 1.596 ms
latency stddev = 0.583 ms
initial connection time = 389.658 ms
tps = 12492.696801 (without initial connection time)
statement latencies in milliseconds and failures:
  0.001      0  \set aid random(1, 10000 * :scale)
  0.000      0  \set bid random(1, 1 * :scale)
  0.000      0  \set tid random(1, 10 * :scale)
  0.000      0  \set delta random(-5000, 5000)
  0.109      0  BEGIN;
  0.191      0  UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
  0.167      0  SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
  0.252      0  UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
  0.457      0  UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
  0.150      0  INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);
  0.268      0  END;
```

Рисунок 15 - Результаты теста в конфигурации XMP-3200
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.16>

По результатам тестирования конфигурации XMP-3200 основные метрики показали 12492 транзакции в секунду и среднюю задержку в 1,596 миллисекунд (рисунок 16).

```
C:\Users\Michael Suyagin>pgbench -c 20 -j 4 -T 60 -P 10 -r -U postgres -d diploma
Password:
pgbench (17.2)
starting vacuum...end.
progress: 10.0 s, 11274.7 tps, lat 1.699 ms stddev 0.998, 0 failed
progress: 20.0 s, 12265.4 tps, lat 1.625 ms stddev 0.676, 0 failed
progress: 30.0 s, 11652.0 tps, lat 1.711 ms stddev 0.813, 0 failed
progress: 40.0 s, 11244.1 tps, lat 1.773 ms stddev 0.892, 0 failed
progress: 50.0 s, 11945.5 tps, lat 1.669 ms stddev 0.749, 0 failed
progress: 60.0 s, 12407.8 tps, lat 1.607 ms stddev 0.628, 0 failed
transaction type: <builtin: TPC-B (sort of)>
scaling factor: 10
query mode: simple
number of clients: 20
number of threads: 4
maximum number of tries: 1
duration: 60 s
number of transactions actually processed: 707913
number of failed transactions: 0 (0.000%)
latency average = 1.679 ms
latency stddev = 0.800 ms
initial connection time = 395.468 ms
tps = 11876.333651 (without initial connection time)
statement latencies in milliseconds and failures:
  0.001      0  \set aid random(1, 10000 * :scale)
  0.000      0  \set bid random(1, 1 * :scale)
  0.000      0  \set tid random(1, 10 * :scale)
  0.000      0  \set delta random(-5000, 5000)
  0.108      0  BEGIN;
  0.206      0  UPDATE pgbench_accounts SET abalance = abalance + :delta WHERE aid = :aid;
  0.169      0  SELECT abalance FROM pgbench_accounts WHERE aid = :aid;
  0.280      0  UPDATE pgbench_tellers SET tbalance = tbalance + :delta WHERE tid = :tid;
  0.494      0  UPDATE pgbench_branches SET bbalance = bbalance + :delta WHERE bid = :bid;
  0.152      0  INSERT INTO pgbench_history (tid, bid, aid, delta, mtime) VALUES (:tid, :bid, :aid, :delta, CURRENT_TIMESTAMP);
  0.289      0  END;
```

Рисунок 16 - Результаты теста в конфигурации 1333 МГц
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.17>

По результатам тестирования в конфигурации 1333МГц наблюдается снижение производительности. Таким образом, количество транзакций в секунду составляет 11876, а средняя задержка выросла до 1,679 миллисекунд.

Для дальнейшего тестирования был написан SQL-запрос (рисунок 17), который структурно построен на последовательном применении обобщенных табличных выражений (англ. Common Table Expressions, CTE) для разбиения задачи на логические этапы, где каждый из них предъявляет высокие требования к оперативной памяти. Основная нагрузка возникает при выполнении JOIN операции между таблицами orders, order_items, products, categories и suppliers.

Агрегации в CTE user_stats, product_stats и category_stats требуют создания хэш-таблиц для группировки данных по пользователям, товарам и категориям. Подобного рода операции требуют значительный объём памяти, особенно при подсчёте уникальных значений (COUNT DISTINCT) и вычислении оконных функций ROW_NUMBER, которые сортируют данные по выручке (рисунок 18).

```
37     SUM(line_total) AS total_revenue,
38     AVG(line_total) AS avg_revenue_per_order,
39     ROW_NUMBER() OVER (ORDER BY SUM(line_total) DESC) AS popularity_rank
40   FROM detailed_orders
41   GROUP BY product_id, product_name
42 ),
43
44 category_stats AS (
45   SELECT
46     Category_name,
47     COUNT(DISTINCT product_id) AS num_products,
48     SUM(line_total) AS total_category_revenue
49   FROM detailed_orders
50   GROUP BY category_name
51 )
52
53
54   SELECT
55     us.full_name,
56     us.order_count,
57     us.total_spent,
58     ps.product_name,
59     ps.total_revenue,
60     ps.popularity_rank,
61     cs.category_name,
62     cs.total_category_revenue
63   FROM user_stats us
64   JOIN detailed_orders d ON d.user_id = us.user_id
65   JOIN product_stats ps ON ps.product_id = d.product_id
66   JOIN category_stats cs ON cs.category_name = d.category_name
67   ORDER BY us.total_spent DESC, ps.popularity_rank ASC
68   LIMIT 100;
```

Data Output Messages Notifications

	full_name	order_count	total_spent	product_name	total_revenue	popularity_rank	category_name	total_category_revenue
1	Timothy Smith	8	82077.70	weight	12149.51	25398	forget	2755954.86
2	Timothy Smith	8	82077.70	Congress	11855.64	27851	to	2756153.57
3	Timothy Smith	8	82077.70	bank	10279.80	45807	enter	3641423.24
4	Timothy Smith	8	82077.70	million	9934.17	51055	very	2687562.41

Total rows: 100 Query complete 00:00:09.190

Рисунок 17 - Результат выполнения запроса в конфигурации памяти XMP-3200

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.18>

Следует отметить, что изменение количества возвращаемых строк с помощью оператора LIMIT не оказывает существенного влияния на общее время выполнения запроса. Это обусловлено тем, что указанный оператор применяется на финальной стадии выполнения SQL-запроса — после того, как все ресурсоёмкие операции, такие как сканирование таблиц, соединение (JOIN), фильтрация (WHERE), группировка (GROUP BY) и сортировка (ORDER BY), уже были выполнены. Таким образом, ограничение выборки не приводит к уменьшению объёма вычислений, необходимых для формирования результирующего набора данных.

Это особенно важно учитывать при проектировании и оптимизации запросов, поскольку простое добавление LIMIT без должной оптимизации самой структуры запроса может создать ложное ощущение его эффективности.

Query _ Query History

```
1 WITH detailed_orders AS (
2     SELECT
3         o.id AS order_id,
4         o.order_date,
5         o.user_id,
6         oi.product_id,
7         oi.quantity,
8         oi.price,
9         p.name AS product_name,
10        c.name AS category_name,
11        s.name AS supplier_name,
12        (oi.quantity * oi.price) AS line_total
13    FROM orders o
14    JOIN order_items oi ON oi.order_id = o.id
15    JOIN products p ON p.id = oi.product_id
16    JOIN categories c ON c.id = p.category_id
17    JOIN suppliers s ON s.id = p.supplier_id
18 ),
19
20 user_stats AS (
21     SELECT
22         d.user_id,
23         u.first_name || ' ' || u.last_name AS full_name,
24         COUNT(DISTINCT d.order_id) AS order_count,
25         SUM(d.line_total) AS total_spent,
26         AVG(d.line_total) AS avg_line_total
27     FROM detailed_orders d
28     JOIN users u ON u.id = d.user_id
29     GROUP BY d.user_id, full_name
30 ),
31
```

Data Output Messages Notifications

Showing rows: 1 t

	full_name	order_count	total_spent	product_name	total_revenue	popularity_rank	category_name	total_category_revenue
	text	bigint	numeric	character varying (100)	numeric	bigint	character varying (100)	numeric
1	Timothy Smith	8	82077.70	weight	12149.51	25398	forget	2755954.86
2	Timothy Smith	8	82077.70	Congress	11855.64	27851	to	2756153.57
3	Timothy Smith	8	82077.70	bank	10279.80	45807	enter	3641423.24
4	Timothy Smith	8	82077.70	million	9934.17	51055	very	2687562.41

Total rows: 1000 Query complete 00:00:10.566

Рисунок 18 - Результат выполнения запроса в конфигурации памяти 1333MHz

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.19>

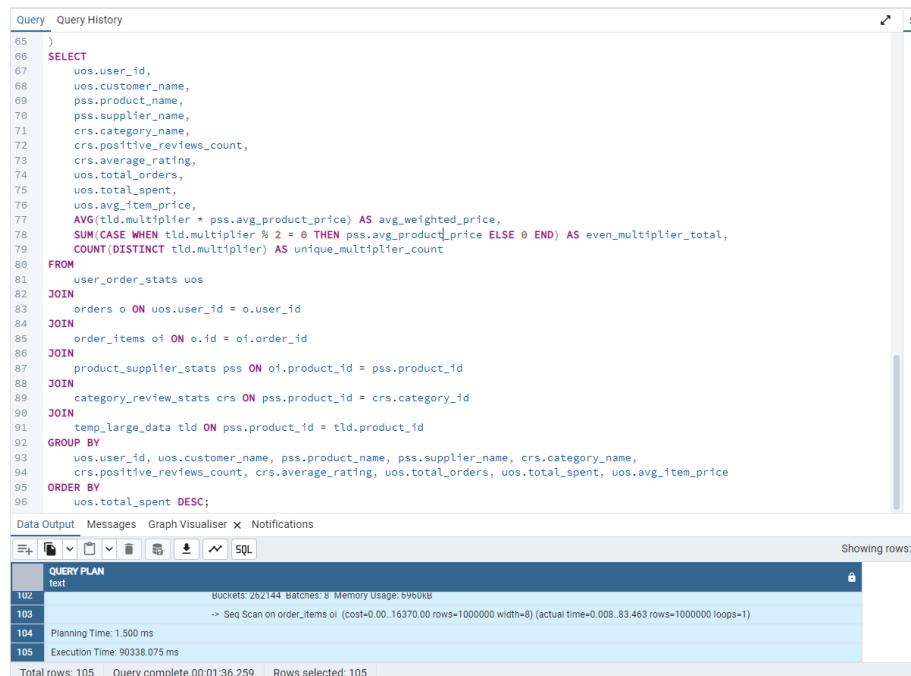
По результатам тестирования производительности с помощью первого запроса наблюдается крайне незначительная разница в скорости выполнения в разных конфигурациях.

При исполнении запроса также можно воспользоваться встроенным инструментом EXPLAIN ANALYZE, который может показать не только план выполнения запроса, но и реальное время выполнения в миллисекундах, количество обработанных строк, количество итераций процесса, время оптимизации запроса.

Далее для сравнения конфигураций в условиях повышенной нагрузки был написан запрос с еще более сложной структурой, который представляет собой, как и CTE, так и объединение таблиц, агрегацию и генерацию данных.

Наиболее ресурсоёмким является использование функции generate_series (1, 100) в CTE temp_large_data, так как она приводит к увеличению числа строк в 100 раз относительно исходных данных в таблице products.

Помимо этого, использование оператора COUNT(DISTINCT ...) в нескольких CTE и в основном запросе приводит к подсчёту уникальных значений и созданию временных данных, таких как хэш-таблицы или деревья (рисунок 19).



```

Query  Query History
65  )
66  SELECT
67    uos.user_id,
68    uos.customer_name,
69    pss.product_name,
70    pss.supplier_name,
71    crs.category_name,
72    crs.positive_reviews_count,
73    crs.average_rating,
74    uos.total_orders,
75    uos.total_spent,
76    uos.avg_item_price,
77    AVG(tld.multiplier * pss.avg_product_price) AS avg_weighted_price,
78    SUM(CASE WHEN tld.multiplier % 2 = 0 THEN pss.avg_product_price ELSE 0 END) AS even_multiplier_total,
79    COUNT(DISTINCT tld.multiplier) AS unique_multiplier_count
80  FROM
81    user_order_stats uos
82  JOIN
83    orders o ON uos.user_id = o.user_id
84  JOIN
85    order_items oi ON o.id = oi.order_id
86  JOIN
87    product_supplier_stats pss ON oi.product_id = pss.product_id
88  JOIN
89    category_review_stats crs ON pss.product_id = crs.category_id
90  JOIN
91    temp_large_data tld ON pss.product_id = tld.product_id
92  GROUP BY
93    uos.user_id, uos.customer_name, pss.product_name, pss.supplier_name, crs.category_name,
94    crs.positive_reviews_count, crs.average_rating, uos.total_orders, uos.total_spent, uos.avg_item_price
95  ORDER BY
96    uos.total_spent DESC;
Data Output  Messages  Graph Visualiser  Notifications
Showing rows: 105
QUERY PLAN
text
Buckets: 262144  Batches: 8  Memory Usage: 6960KB
-> Seq Scan on order_items oi  (cost=0.00..16370.00 rows=1000000 width=8) (actual time=0.008..83.463 rows=1000000 loops=1)
102
103
104 Planning Time: 1.500 ms
105 Execution Time: 90338.075 ms
Total rows: 105  Query complete 00:01:36.259  Rows selected: 105

```

Рисунок 19 - Результат выполнения второго запроса в конфигурации памяти XMP-3200

DOI: <https://doi.org/10.60797/IRJ.2025.161.48.20>

На рисунках 21 и 22 показаны результат выполнения второго запроса в конфигурации памяти 1333MHz и время выполнения запроса с использованием инструмента EXPLAIN ANALYSE во второй конфигурации соответственно.

```

Query 1
  c.id AS category_id,
  c.name AS category_name,
  COUNT(r.rating) FILTER (WHERE r.rating >= 4) AS positive_reviews_count,
  AVG(r.rating) AS average_rating
FROM
  categories c
JOIN
  products p ON c.id = p.category_id
LEFT JOIN
  reviews r ON p.id = r.product_id
GROUP BY
  c.id, c.name
),
temp_large_data AS (
  -- Временная таблица с увеличенными данными
  SELECT
    p.id AS product_id,
    p.name AS product_name,
    c.name AS category_name,
    generate_series(1, 100) AS multiplier
  FROM
    products p
  JOIN
    categories c ON p.category_id = c.id
)
-- Финальный запрос
SELECT
  uos.user_id,
  uos.customer_name,
  pss.product_name,
  pss.supplier_name,
  crs.category_name,
  crs.positive_reviews_count
FROM
  temp_large_data AS (
    SELECT
      uos.user_id,
      uos.customer_name,
      pss.product_name,
      pss.supplier_name,
      crs.category_name,
      crs.positive_reviews_count
    FROM
      users uos
    JOIN
      products pss ON uos.user_id = pss.user_id
    JOIN
      categories crs ON pss.category_id = crs.id
  )
  
```

Showing rows: Lock Refresh Stop

QUERY PLAN

```

1  GroupAggregate (cost=1781420.39..1797553.61 rows=268887 width=249) (actual time=93439.240..98982.003 rows=49974 loops=1)
2  Group Key: uos.total_spent, uos.user_id, uos.customer_name, pss.product_name, pss.supplier_name, crs.category_name, crs.positive_reviews_count, uos.average_rating, uos.total_orders, uos.a...
3  -> Sort (cost=1781420.39..1782092.60 rows=268887 width=213) (actual time=93439.11..95049.226 rows=4997400 loops=1)
4  -> Sort Key: uos.total_spent DESC, uos.user_id, uos.customer_name, pss.product_name, pss.supplier_name, crs.category_name, crs.positive_reviews_count, crs.average_rating, uos.total_order...
Total rows: 105  Query complete 00:01:41.047
  
```

Рисунок 20 - Результат выполнения второго запроса в конфигурации памяти 1333MHz
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.21>

104	Planning Time: 25.776 ms
105	Execution Time: 99060.980 ms
Total rows: 105 Query complete 00:01:41.047	

Рисунок 21 - Время выполнения запроса с использованием инструмента EXPLAIN ANALYSE во второй конфигурации
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.22>

Анализ результатов, полученных в ходе проведённого исследования влияния характеристик оперативной памяти DDR4 на производительность PostgreSQL, выявил ряд значимых тенденций, подкреплённых как статистической обработкой данных, так и сопоставлением с результатами аналогичных исследований.

В рамках тестирования были зафиксированы ключевые показатели производительности системы при двух различных конфигурациях оперативной памяти: XMP-3200 и пониженной 1333 МГц. В частности, производительность, выраженная в количестве транзакций в секунду (TPS), составила 12492 для конфигурации XMP-3200 и 11876 для конфигурации 1333 МГц. Средняя задержка обработки запроса в первом случае равнялась 1,596 мс, тогда как во втором увеличивалась до 1,679 мс. Разница в TPS составила около 4,9%, а рост задержек — порядка 5,2%.

Для подтверждения статистической значимости этих различий применялись стандартные методы анализа, включая расчёт относительных изменений, доверительных интервалов и стандартного отклонения. Повторные измерения позволили получить репрезентативные выборки, на основе которых можно было провести t-тест Стьюдента. При уровне значимости 0,05 различия в производительности оказались статистически значимыми, что указывает на достоверное влияние изменения частоты оперативной памяти на производительность PostgreSQL.

Также наблюдалась тенденция увеличения времени выполнения ресурсоёмких SQL-запросов в условиях пониженной пропускной способности и возросших таймингов. Запросы с использованием оконных функций, агрегаций и генерации массивов данных с помощью `generate_series` показали чувствительность к изменениям в латентности и пропускной способности памяти. В случае конфигурации 1333 МГц фиксировалось увеличение времени обработки сложных выборок, несмотря на то, что общая логика выполнения запросов оставалась неизменной.

Сопоставление с результатами других исследований показало, что полученные данные находятся в рамках существующих научных представлений о влиянии оперативной памяти на производительность СУБД. Так, исследования [25], [26], [27] демонстрируют, что при OLTP-нагрузке рост частоты ОЗУ приводит к увеличению производительности на 5-7%, что сопоставимо с данными настоящей работы. Аналогичные выводы сделаны в работе [28], [29], [30], где указано, что для PostgreSQL особенно критична пропускная способность при выполнении операций JOIN и агрегации.

Однако стоит учитывать, что по сравнению с in-memory СУБД, таких как Redis или MemSQL, PostgreSQL в меньшей степени зависит от задержек доступа к памяти, поскольку использует собственные механизмы кэширования и оптимизации выполнения запросов. В связи с этим влияние латентности, обусловленной таймингами памяти, оказалось менее значимым, чем влияние частоты и пропускной способности.

Таким образом, статистическая обработка результатов и их сопоставление с данными других авторов позволяют заключить, что увеличение частоты оперативной памяти может дать ощутимый прирост производительности PostgreSQL, особенно в сценариях с интенсивной нагрузкой на память. Тем не менее, для систем со средней или низкой нагрузкой, разница между конфигурациями может оказаться недостаточно весомой для оправдания повышения затрат на более дорогую память. Такой подход обеспечивает обоснованность и объективность сделанных выводов, позволяя применять их в практике проектирования и модернизации серверов баз данных.

Заключение

Проведённое исследование позволило дать количественно обоснованную оценку влияния характеристик оперативной памяти DDR4 на производительность системы управления базами данных PostgreSQL. Полученные данные свидетельствуют о том, что тактовая частота памяти оказывает непосредственное влияние на основные метрики производительности СУБД, включая количество транзакций в секунду (TPS) и среднюю задержку выполнения запросов. При снижении частоты памяти с 3200 МГц до 1333 МГц наблюдалось уменьшение производительности на 4,9% и увеличение средней задержки на 5,2%, что статистически подтверждено многоократными измерениями и анализом данных.

Тайминги памяти, выраженные в тактах, оказались менее критичными показателями при сравнении двух режимов, особенно с учётом пересчёта в наносекунды, где более высокочастотная память демонстрировала меньшие абсолютные значения задержек. Это позволяет утверждать, что в условиях PostgreSQL, активно использующей кэширование и внутреннюю оптимизацию выполнения запросов, решающее значение имеет именно пропускная способность оперативной памяти, а не её латентность.

Анализ SQL-запросов, включающих оконные функции, агрегации и операции соединения (JOIN), показал, что увеличение частоты памяти положительно сказывается на времени их выполнения, особенно при работе с большими объёмами данных. Таким образом, для OLTP-систем и аналитических платформ, ориентированных на интенсивную память-нагрузку, инвестиции в более высокочастотную память могут быть оправданы. В то же время для серверов с умеренной или предсказуемой нагрузкой разница может быть нивелирована за счёт настройки параметров PostgreSQL и системного кэширования.

Возможные направления дальнейших исследований:

1. Сравнение с другими поколениями памяти, включая DDR5, с целью выявления не только преимуществ по производительности, но и их энергоэффективности и экономической целесообразности в контексте современных СУБД.

2. Изучение зависимости между количеством каналов памяти (single vs. dual/quad channel) и производительностью PostgreSQL в многопользовательской среде.

3. Оценка влияния архитектурных особенностей процессора, включая размер кэшей L2/L3 и поддержку SMT (Simultaneous Multithreading), на чувствительность к частоте оперативной памяти при работе СУБД.

4. Разработка интеллектуальных систем автоматической настройки PostgreSQL, учитывающих текущую конфигурацию памяти и тип нагрузки, что позволит обеспечить оптимальную производительность без участия администратора.

5. Анализ поведения PostgreSQL в условиях ограниченного ресурса ОЗУ, включая сценарии с перегрузкой памяти, выходом за пределы shared_buffers и интенсивным использованием swap, что особенно актуально для облачных решений и контейнеризованных сред.

Таким образом, данное исследование создаёт прочную основу для дальнейших эмпирических и теоретических разработок, направленных на оптимизацию аппаратных и программных компонентов серверных решений, использующих PostgreSQL в качестве основной СУБД.

Конфликт интересов

Не указан.

Рецензия

Черняков М.К., Сибирский университет потребительской кооперации, Новосибирск Российская Федерация
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.23>

Conflict of Interest

None declared.

Review

Chernyakov M.K., Siberian University of Consumer Cooperatives, Novosibirsk Russian Federation
DOI: <https://doi.org/10.60797/IRJ.2025.161.48.23>

Список литературы / References

1. Kumar P. Design and Verification of DDR SDRAM Memory Controller Using SystemVerilog For Higher Coverage / P. Kumar, S.K. Panda // 2019 International Conference on Intelligent Computing and Control Systems (ICCS). — 2019. — P. 689–694. — DOI: 10.1109/ICCS45141.2019.9065407.
2. Bakshi A. ASIC implementation of DDR SDRAM Memory Controller / A. Bakshi, S.S. Pandey, T. Pradhan, R. Dey // 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN). — 2013. — P. 74–78. — DOI: 10.1109/ICE-CCN.2013.6528467.
3. Екубдjonов Д.И. Исследование производительности технологий объектно-реляционного отображения при взаимодействии с Microsoft SQL Server / Д.И. Екубдjonов, Р.Ф. Гибадуллин // Международный научно-исследовательский журнал. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.145.

4. Gibadullin R.F. Realization of replication mechanism in PostgreSQL DBMS / R.F. Gibadullin, I.S. Vershinin, R.S. Minyazev // 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). — 2017. — P. 1–6. — DOI: 10.1109/ICIEAM.2017.8076380.
5. Mathew B.K. Design of a parallel vector access unit for SDRAM memory systems / B.K. Mathew, S.A. McKee, J.B. Carter, A. Davis // Proceedings Sixth International Symposium on High-Performance Computer Architecture. HPCA-6. — 2000. — P. 39–48. — DOI: 10.1109/HPCA.2000.824337.
6. Vaslavskaya I. The Use of Blockchain Technology for Transport and Logistics Systems in the Digital Economy / I. Vaslavskaya, I. Koshkina, R. Zaripova // Finance, Economics, and Industry for Sustainable Development. — Cham: Springer, 2024. — P. 189–201. — DOI: 10.1007/978-3-031-56380-5_16.
7. Гилемханов Т.Ф. Автоматизированный стенд для измерения параметров источников электропитания / Т.Ф. Гилемханов, Р.Ф. Гибадуллин // Международный научно-исследовательский журнал. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.70.
8. Шигабетдинова Д.И. Обнаружение и локализация утечек на нефтедобывающих объектах с помощью компьютерного зрения / Д.И. Шигабетдинова, З.М. Гизатуллин, М.П. Шлеймович // Вестник Технологического университета. — 2025. — Т. 28. — № 5. — С. 123–128. — DOI: 10.55421/3034-4689_2025_28_5_123.
9. Pandey A.K. Signal and power integrity analysis of DDR4 address bus of onboard memory module / A.K. Pandey // 2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS). — 2018. — P. 1–3. — DOI: 10.1109/EDAPS.2018.8680896.
10. Шарипов Р.Р. Разработка программного комплекса потокового шифра RC4 для обучающихся по дисциплине «криптография» / Р.Р. Шарипов, С.П. Макаров, А.А. Кассирова // Международный научно-исследовательский журнал. — 2024. — № 9 (147). — DOI: 10.60797/IRJ.2024.147.15.
11. Yang S.-W. Structural demonstration of cost effective isolation trench fill for sub-110 nm vertical trench DRAM and SOC applications / S.-W. Yang [et al.] // 2003 International Symposium on VLSI Technology, Systems and Applications. — 2003. — P. 117–120. — DOI: 10.1109/VTSA.2003.1252566.
12. Шкиндеров М.С. Моделирование помехоустойчивости системы контроля и управления доступом при воздействии электростатического разряда / М.С. Шкиндеров, Р.Р. Мубараков // Труды МАИ. — 2021. — № 120. — С. 12–20. — DOI: 10.34759/trd-2021-120-12.
13. Guet F. Probabilistic analysis of cache memories and cache memories impacts on multi-core embedded systems / F. Guet, L. Santinelli, J. Morio // 2016 11th IEEE Symposium on Industrial Embedded Systems (SIES). — 2016. — P. 1–10. — DOI: 10.1109/SIES.2016.7509420.
14. Шурдилов И.С. Разработка системы распознавания эмоций по лицевым выражениям на основе машинного обучения / И.С. Шурдилов, М.Г. Нуриев, М.Г. Лаптева [и др.] // Международный научно-исследовательский журнал. — 2025. — № 6 (156). — С. 52–58. — DOI: 10.60797/IRJ.2025.156.52.
15. Yin H. Optimization of WinRAR Password Cracking Algorithm Based on Heterogeneous Computing / H. Yin, L. Ni // 2021 IEEE 21st International Conference on Communication Technology (ICCT). — 2021. — P. 892–896. — DOI: 10.1109/ICCT52962.2021.9658021.
16. Катасёв А.С. Нейронечеткая модель и программный комплекс автоматизации формирования нечетких правил для оценки состояния объектов / А.С. Катасёв // Автоматизация процессов управления. — 2019. — № 1 (55). — С. 21–29.
17. Machado R.R. UEFI BIOS Accessibility for the Visually Impaired / R.R. Machado, G.M.D. Vieira // 2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC). — 2017. — P. 155–160. — DOI: 10.1109/SBESC.2017.27.
18. Шакирзянов Р.М. Метод автоматического позиционирования беспилотных аппаратов на основе распознавания сигнальных радиально-симметричных маркеров подводных целей / Р.М. Шакирзянов, М.П. Шлеймович, С.В. Новикова // Автоматика и телемеханика. — 2023. — № 7. — С. 93–120. — DOI: 10.31857/S0005231023070061.
19. Yerramilli N.S. College Exam Allocation Using MongoDB and Python3 / N.S. Yerramilli, N.J. Johnson, Y. Omsri Sainadh Reddy // 2021 2nd Global Conference for Advancement in Technology (GCAT). — 2021. — P. 1–3. — DOI: 10.1109/GCAT52182.2021.9587589.
20. Смирнов Ю.Н. Математическая модель оптимизации деятельности для цифровой системы управления предприятием / Ю.Н. Смирнов, А.В. Каляшина // Научно-технический вестник Поволжья. — 2023. — № 11. — С. 119–122.
21. Suresh babu C.V. Web-Based Deep Learning Model for Zero Day Vulnerability Detection using FastAPI / C.V. Suresh babu, V. Surendar, E. Sriram, S. Subhash // 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). — 2024. — P. 1–6. — DOI: 10.1109/ADICS58448.2024.10533540.
22. Хабибуллин Ф.Ф. Анализ параметров ошибки положения, перемещения идеального и реального механизма для робототехнических систем / Ф.Ф. Хабибуллин, Р.Т. Исламов, Л.Ф. Хабибуллина // Проблемы машиностроения и автоматизации. — 2024. — № 1. — С. 36–43.
23. White K.A. Single-Cell Recording of Vesicle Release From Human Neuroblastoma Cells Using 1024-ch Monolithic CMOS Bioelectronics / K.A. White [et al.] // IEEE Transactions on Biomedical Circuits and Systems. — 2018. — Vol. 12. — № 6. — P. 1345–1355. — DOI: 10.1109/TBCAS.2018.2861220.
24. Brigida V. Technogenic Reservoirs Resources of Mine Methane When Implementing the Circular Waste Management Concept / V. Brigida, V.I. Golik, E.V. Voitovich [et al.] // Resources. — 2024. — Vol. 13, № 2. — Art. 33. — DOI: 10.3390/resources13020033.

25. Beigi M.V. A systematic study of ddr4 dram faults in the field / M.V. Beigi [et al.] // 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). — 2023. — P. 991–1002. — DOI: 10.1109/HPCA56546.2023.10071021.
26. Zhou R. A Novel Insight Into the Vulnerability of DDR4 DRAM Cells Across Multiple Hammering Settings / R. Zhou [et al.] // IEEE Embedded Systems Letters. — 2024. — Vol. 16. — № 4. — P. 337–340. — DOI: 10.1109/LES.2023.3327590.
27. Priyanka B. High Coverage DDR4 SDRAM Memory Design and Verification Using System Verilog and UVM / B. Priyanka, B.A. Kumar, D.S. Kumar // 2024 2nd International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV). — 2024. — P. 1–6. — DOI: 10.1109/ICPEEV61244.2024.10833645.
28. Beigi M.V. DDR5 DRAM Faults in the Field / M.V. Beigi [et al.] // 2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). — 2025. — P. 36–41. — DOI: 10.1109/DSN-S63370.2025.00016.
29. Von Thun M. SEU and SEFI Characterization of a Frontgrade 18GB DDR4 Memory for Space Applications / M. Von Thun [et al.] // 2024 IEEE Radiation Effects Data Workshop (REDW). — 2024. — P. 1–4. — DOI: 10.1109/REDW63039.2024.10796515.
30. Mutlu O. Memory-Centric Computing: Solving Computing's Memory Problem / O. Mutlu, A. Olgun, İ.E. Yüksel // 2025 IEEE International Memory Workshop (IMW). — 2025. — P. 1–4. — DOI: 10.1109/IMW62431.2025.10573562.

Список литературы на английском языке / References in English

1. Kumar P. Design and Verification of DDR SDRAM Memory Controller Using SystemVerilog For Higher Coverage / P. Kumar, S.K. Panda // 2019 International Conference on Intelligent Computing and Control Systems (ICCS). — 2019. — P. 689–694. — DOI: 10.1109/ICCS45141.2019.9065407.
2. Bakshi A. ASIC implementation of DDR SDRAM Memory Controller / A. Bakshi, S.S. Pandey, T. Pradhan, R. Dey // 2013 IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN). — 2013. — P. 74–78. — DOI: 10.1109/ICE-CCN.2013.6528467.
3. Ekjubdzhonov D.I. Issledovanie proizvoditel'nosti tehnologij ob"ektno-reljacionnogo otobrazhenija pri vzaimodejstvii s Microsoft SQL Server [Performance study of object-relational mapping technologies when interacting with Microsoft SQL Server] / D.I. Ekjubdzhonov, R.F. Gibadullin // Mezhdunarodnyj nauchno-issledovatel'skij zhurnal [International Research Journal]. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.145. [in Russian]
4. Gibadullin R.F. Realization of replication mechanism in PostgreSQL DBMS / R.F. Gibadullin, I.S. Vershinin, R.S. Minyazev // 2017 International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM). — 2017. — P. 1–6. — DOI: 10.1109/ICIEAM.2017.8076380.
5. Mathew B.K. Design of a parallel vector access unit for SDRAM memory systems / B.K. Mathew, S.A. McKee, J.B. Carter, A. Davis // Proceedings Sixth International Symposium on High-Performance Computer Architecture. HPCA-6. — 2000. — P. 39–48. — DOI: 10.1109/HPCA.2000.824337.
6. Vaslavskaya I. The Use of Blockchain Technology for Transport and Logistics Systems in the Digital Economy / I. Vaslavskaya, I. Koshkina, R. Zaripova // Finance, Economics, and Industry for Sustainable Development. — Cham: Springer, 2024. — P. 189–201. — DOI: 10.1007/978-3-031-56380-5_16.
7. Gilemhanov T.F. Avtomatizirovannyj stend dlja izmerenija parametrov istochnikov jeklropitanija [Automated test bench for measuring power supply parameters] / T.F. Gilemhanov, R.F. Gibadullin // Mezhdunarodnyj nauchno-issledovatel'skij zhurnal [International Research Journal]. — 2024. — № 10 (148). — DOI: 10.60797/IRJ.2024.148.70. [in Russian]
8. Shigabetdinova D.I. Obnaruzhenie i lokalizacija utechek na neftedobyvajushhih ob"ektah s pomoshh'ju komp'juternogo zrenija [Detection and localization of leaks at oil production facilities using computer vision] / D.I. Shigabetdinova, Z.M. Gizatullin, M.P. Shlejmovich // Vestnik Tehnologicheskogo universiteta [Bulletin of the Technological University]. — 2025. — Vol. 28. — № 5. — P. 123–128. — DOI: 10.55421/3034-4689_2025_28_5_123. [in Russian]
9. Pandey A.K. Signal and power integrity analysis of DDR4 address bus of onboard memory module / A.K. Pandey // 2018 IEEE Electrical Design of Advanced Packaging and Systems Symposium (EDAPS). — 2018. — P. 1–3. — DOI: 10.1109/EDAPS.2018.8680896.
10. Sharipov R.R. Razrabotka programmnogo kompleksa potochnogo shifra RC4 dlja obuchajushhihsja po discipline «kriptografija» [Development of a software complex for the RC4 stream cipher for students in the "cryptography" discipline] / R.R. Sharipov, S.P. Makarov, A.A. Kassirova // Mezhdunarodnyj nauchno-issledovatel'skij zhurnal [International Research Journal]. — 2024. — № 9 (147). — DOI: 10.60797/IRJ.2024.147.15. [in Russian]
11. Yang S.-W. Structural demonstration of cost effective isolation trench fill for sub-110 nm vertical trench DRAM and SOC applications / S.-W. Yang [et al.] // 2003 International Symposium on VLSI Technology, Systems and Applications. — 2003. — P. 117–120. — DOI: 10.1109/VTSA.2003.1252566.
12. Shkinderov M.S. Modelirovanie pomehoustojchivosti sistemy kontrolja i upravlenija dostupom pri vozdejstvii jeklektrostaticheskogo razrjada [Modeling the noise immunity of an access control system under electrostatic discharge] / M.S. Shkinderov, R.R. Mubarakov // Trudy MAI [Proceedings of MAI]. — 2021. — № 120. — P. 12–20. — DOI: 10.34759/trd-2021-120-12. [in Russian]
13. Guet F. Probabilistic analysis of cache memories and cache memories impacts on multi-core embedded systems / F. Guet, L. Santinelli, J. Morio // 2016 11th IEEE Symposium on Industrial Embedded Systems (SIES). — 2016. — P. 1–10. — DOI: 10.1109/SIES.2016.7509420.
14. Shurdilov I.S. Razrabotka sistemy raspoznavaniija jemocij po licem v yrazhenijam na osnove mashinnogo obuchenija [Development of a facial expression emotion recognition system based on machine learning] / I.S. Shurdilov, M.G. Nuriev,

- M.G. Lapteva [et al.] // Mezhdunarodnyj nauchno-issledovatel'skij zhurnal [International Research Journal]. — 2025. — № 6 (156). — P. 52–58. — DOI: 10.60797/IRJ.2025.156.52. [in Russian]
15. Yin H. Optimization of WinRAR Password Cracking Algorithm Based on Heterogeneous Computing / H. Yin, L. Ni // 2021 IEEE 21st International Conference on Communication Technology (ICCT). — 2021. — P. 892–896. — DOI: 10.1109/ICCT52962.2021.9658021.
16. Katasjov A.S. Nejronechetkaja model' i programmnyj kompleks avtomatizacii formirovaniya nechetkih pravil dlja ocenki sostojanija ob"ektorov [Neuro-fuzzy model and software complex for automation of fuzzy rule formation for object state assessment] / A.S. Kataljov // Avtomatizacija processov upravlenija [Automation of Control Processes]. — 2019. — № 1 (55). — P. 21–29. [in Russian]
17. Machado R.R. UEFI BIOS Accessibility for the Visually Impaired / R.R. Machado, G.M.D. Vieira // 2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC). — 2017. — P. 155–160. — DOI: 10.1109/SBESC.2017.27.
18. Shakirzjanov R.M. Metod avtomaticheskogo pozicionirovaniya bespilotnyh apparatov na osnove raspoznavaniya signal'nyh radial'no-simmetrichnyh markerov podvodnyh celej [Method for automatic positioning of unmanned vehicles based on recognition of signal radially symmetric markers of underwater targets] / R.M. Shakirzjanov, M.P. Shlejmovich, S.V. Novikova // Avtomatika i telemekhanika [Automation and Remote Control]. — 2023. — № 7. — P. 93–120. — DOI: 10.31857/S0005231023070061. [in Russian]
19. Yerramilli N.S. College Exam Allocation Using MongoDB and Python3 / N.S. Yerramilli, N.J. Johnson, Y. Omsri Sainadh Reddy // 2021 2nd Global Conference for Advancement in Technology (GCAT). — 2021. — P. 1–3. — DOI: 10.1109/GCAT52182.2021.9587589.
20. Smirnov Ju.N. Matematicheskaja model' optimizacii dejatel'nosti dlja cifrovoj sistemy upravlenija predpriyatiem [Mathematical model of activity optimization for a digital enterprise management system] / Ju.N. Smirnov, A.V. Kaljashina // Nauchno-tehnicheskij vestnik Povolzh'ja [Scientific and Technical Bulletin of the Volga Region]. — 2023. — № 11. — P. 119–122. [in Russian]
21. Suresh babu C.V. Web-Based Deep Learning Model for Zero Day Vulnerability Detection using FastAPI / C.V. Suresh babu, V. Surendar, E. Sriram, S. Subhash // 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS). — 2024. — P. 1–6. — DOI: 10.1109/ADICS58448.2024.10533540.
22. Habibullin F.F. Analiz parametrov oshibki polozhenija, peremeshchenija ideal'nogo i real'nogo mehanizma dlja robototekhnicheskikh sistem [Analysis of position error parameters, displacement of ideal and real mechanism for robotic systems] / F.F. Habibullin, R.T. Islamov, L.F. Habibullina // Problemy mashinostroenija i avtomatizacii [Problems of Mechanical Engineering and Automation]. — 2024. — № 1. — P. 36–43. [in Russian]
23. White K.A. Single-Cell Recording of Vesicle Release From Human Neuroblastoma Cells Using 1024-ch Monolithic CMOS Bioelectronics / K.A. White [et al.] // IEEE Transactions on Biomedical Circuits and Systems. — 2018. — Vol. 12. — № 6. — P. 1345–1355. — DOI: 10.1109/TBCAS.2018.2861220.
24. Brigida V. Technogenic Reservoirs Resources of Mine Methane When Implementing the Circular Waste Management Concept / V. Brigida, V.I. Golik, E.V. Voitovich [et al.] // Resources. — 2024. — Vol. 13, № 2. — Art. 33. — DOI: 10.3390/resources13020033.
25. Beigi M.V. A systematic study of ddr4 dram faults in the field / M.V. Beigi [et al.] // 2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA). — 2023. — P. 991–1002. — DOI: 10.1109/HPCA56546.2023.10071021.
26. Zhou R. A Novel Insight Into the Vulnerability of DDR4 DRAM Cells Across Multiple Hammering Settings / R. Zhou [et al.] // IEEE Embedded Systems Letters. — 2024. — Vol. 16. — № 4. — P. 337–340. — DOI: 10.1109/LES.2023.3327590.
27. Priyanka B. High Coverage DDR4 SDRAM Memory Design and Verification Using System Verilog and UVM / B. Priyanka, B.A. Kumar, D.S. Kumar // 2024 2nd International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV). — 2024. — P. 1–6. — DOI: 10.1109/ICPEEV61244.2024.10833645.
28. Beigi M.V. DDR5 DRAM Faults in the Field / M.V. Beigi [et al.] // 2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks-Supplemental Volume (DSN-S). — 2025. — P. 36–41. — DOI: 10.1109/DSN-S63370.2025.00016.
29. Von Thun M. SEU and SEFI Characterization of a Frontgrade 18GB DDR4 Memory for Space Applications / M. Von Thun [et al.] // 2024 IEEE Radiation Effects Data Workshop (REDW). — 2024. — P. 1–4. — DOI: 10.1109/REDW63039.2024.10796515.
30. Mutlu O. Memory-Centric Computing: Solving Computing's Memory Problem / O. Mutlu, A. Olgun, İ.E. Yüksel // 2025 IEEE International Memory Workshop (IMW). — 2025. — P. 1–4. — DOI: 10.1109/IMW62431.2025.10573562.