

Нагрузочное тестирование информационных систем: виды и методы

Н.И. Какурин, Н.А. Карпов, Д.Д. Руденко

Аннотация—Статья посвящена анализу видов нагрузочного тестирования информационных систем. Представлена классификация видов нагрузочного тестирования. Приведен обзор каждого из представленных видов, описаны возможные области применения, а также даны рекомендации по выбору моделей нагрузки и дополнительных метрик. Перечислены потенциальные сложности, возникающие при формировании профиля нагрузки.

Материал будет полезен специалистам в области тестирования ИС, а также инженерам, чья задача заключается в достижении максимальной эффективности работы информационных систем.

Ключевые слова—нагрузочное тестирование информационных систем, тестирование производительности, классификация видов нагрузочного тестирования, профиль нагрузки.

I. ВВЕДЕНИЕ

Тестирование производительности информационных систем (ИС) играет ключевую роль в обеспечении их надежности, стабильности и эффективности. В современных условиях, требующих высокой доступности, масштабируемости и стрессоустойчивости информационных систем, тестирование производительности становится неотъемлемой частью процесса разработки программного обеспечения (ПО). Одним из важнейших видов тестирования производительности является нагрузочное тестирование (НТ), которое направлено на оценку поведения системы в различных, иногда экстремальных, эксплуатационных условиях.

Нагрузочное тестирование позволяет выявить проблемные зоны в программном обеспечении, такие как узкие места, ошибки конфигурации и недостаточная производительность. Кроме того, оно предоставляет объективные данные о соответствии системы нефункциональным требованиям, например, таким как время отклика системы, утилизация ресурсов серверов и количество ошибок. Отметим, что для достижения максимальной эффективности, нагрузочное тестирование должно проводиться с учетом задач, стоящих перед системой. Кроме того, необходимо четко

определить цели проведения тестирования, и исходя из них выбирать соответствующие виды нагрузочного тестирования.

В целом тестирование производительности может производиться с помощью различных подходов и методов. Методы нагрузочного тестирования, в свою очередь, имеют собственную более узкую классификацию, которая охватывает различные аспекты и описывает разнообразные сценарии проверки системы. В данной статье дается обзор видов нагрузочного тестирования, их сопоставление, а также рекомендации по применению каждого из них в зависимости от конкретных условий и требований.

II. ОБЗОР ЛИТЕРАТУРЫ

Область нагрузочного тестирования информационных систем претерпела значительную эволюцию — от простых измерений времени отклика до сложных методов, интегрированных в жизненный цикл разработки программного обеспечения и управление инфраструктурой. Данный обзор систематизирует ключевые работы и тенденции, сформировавшие современное понимание НТ.

Зарождение методологии нагрузочного тестирования как систематического процесса связано с работами, заложившими теоретический фундамент анализа производительности. Классическим трудом, остающимся актуальным и сегодня, является [2]. В этой работе представлен широкий набор методов и подходов для планирования экспериментов, сбора данных, статистического анализа и интерпретации результатов. Знание методов, описанных в работе [2], таких как анализ очередей, планирование тестов и работа с перцентилями, является обязательным для любого специалиста по производительности.

Непосредственно к практике тестирования приложений относится работа [1], где детально разбирается весь процесс нагрузочного тестирования: от определения целей и метрик до создания сценариев, выполнения тестов и анализа результатов. Вводится четкое различие между типами тестирования и подчеркивается важность связи бизнес-требований и нефункционального требования.

Важнейший вклад в предсказательное моделирование внесли авторы работы [5]. Они предложили математические и аналитические методы прогнозирования производительности системы на основе анализа логов, законов использования и операционного анализа. Их подходы к построению моделей типа "что, если" (what-if models) критически

Статья получена 01.07.2026

Какурин Никита Игоревич, Национальный Исследовательский Ядерный Университет МИФИ, (e-mail: kakurin1999@yandex.ru).

Карпов Никита Алексеевич, Национальный Исследовательский Ядерный Университет МИФИ (e-mail: karpow_nikita@mail.ru).

Руденко Даниил Денисович, Национальный Исследовательский Ядерный Университет МИФИ (e-mail: ddrudenko@ya.ru).

важны для построения адекватных профилей нагрузки, учитывающих поведение пользователей и сезонные колебания.

С распространением интернета и многоуровневых архитектур фокус сместился с тестирования монолитных систем на анализ сложных распределенных сред. Появление сервисно-ориентированных архитектур (SOA), а затем и микросервисов, потребовало новых подходов к мониторингу и анализу.

Исследования, подобные работе [4], обозначали новые вызовы перед инженерами, отвечающими за производительность информационных систем: необходимость распределенного трассирования запросов (distributed tracing) для отслеживания пути запроса через десятки сервисов, сложность управления тестовыми данными в распределенных транзакциях и важность мониторинга не только отдельных узлов, но и интерфейсов.

Работа [3] отражает смещение парадигмы от простой генерации нагрузки к сложному моделированию поведения пользователей. Авторы показывают, как анализ сессий пользователей и сценариев использования позволяет создавать гораздо более точные и репрезентативные профили нагрузки, что приводит к обнаружению реальных узких мест.

Развитие облачных инфраструктур и вычислений породило необходимость тестирования в условиях динамического выделения ресурсов. Работа [6] рассматривает методологии оценки производительности распределенных систем (Cassandra, MongoDB, Elasticsearch, ClickHouse) в гибридных облачных средах, где такие факторы, как сетевая задержка между центрами обработки данных и механизмы автоматического масштабирования, становятся критически важными для производительности.

Эволюция методов неразрывно связана с развитием инструментов, которые прошли путь от простых HTTP-клиентов до мощных платформ.

Apache JMeter долгое время остается де-факто стандартом с открытым исходным кодом благодаря своей мощи и расширяемости. Его основанная на плагинах архитектура позволяет тестировать практически все типы ИС. С другой стороны, Gatling и k6 представляют собой новое поколение инструментов, ориентированных на интеграцию в методологию CI/CD. Их ключевые преимущества — высокая производительность за счет асинхронной архитектуры и нативной поддержки JavaScript, а также возможность включения тестов в кодовую базу (test as code), что обеспечивает лучшую сопровождаемость.

Для тестирования высоконагруженных систем и генерации нагрузки на уровне пакетов используются такие инструменты, как Yandex.Tank и Apache Benchmark. Кроме того, коммерческие платформы типа LoadRunner (Micro Focus) и NeoLoad (Tricentis) предлагают комплексные решения с расширенными возможностями анализа, поддержкой множества протоколов и мощными сервисами управления виртуальными пользователями.

Современный этап развития характеризуется

глубокой интеграцией HT в процессы DevOps, дальнейшей автоматизацией и использованием передовых аналитических методов. В работе [7] описываются методы встраивания автоматического нагрузочного тестирования и профилирования прямо в конвейеры непрерывной поставки для бессерверных архитектур (FaaS), что позволяет обнаруживать регрессии производительности на этапе фиксации кода. Работа [10] демонстрирует, как методы машинного обучения применяются для анализа метрик производительности, выявления аномалий, которые могут быть незаметны человеческому глазу, и даже прогнозирования будущих узких мест и сбоев. Также активно развиваются методики HT для критически важных областей. Например, работа [8] фокусируется на тонкостях тестирования производительности систем больших данных и поисковых движков, где ключевыми являются метрики, связанные с распределением данных, балансировкой нагрузки и эффективностью запросов.

Проведенный анализ литературы показывает, что, несмотря на обилие работ [1-10], посвященных конкретным инструментам и локальным методикам, ощущается недостаток в целостных, систематизированных руководствах, которые бы позволяли осознанно выбирать вид нагрузочного тестирования, адекватный поставленным целям и контексту проекта. В доступной литературе отсутствует единая, общепринятая и всесторонняя классификация видов нагрузочного тестирования или имеющиеся подходы фрагментарны и зависимы от контекста. Данная статья в какой-то мере позволяет восполнить этот пробел, предлагая структурированную и практико-ориентированную классификацию видов и методов HT.

III. КЛАССИФИКАЦИЯ ВИДОВ НАГРУЗОЧНОГО ТЕСТИРОВАНИЯ

Нагрузочное тестирование — это один из видов тестирования производительности, наряду с тестированием надежности и тестированием масштабируемости. Целью проведения HT является изучение поведения системы заданной конфигурации под нагрузками.

В свою очередь, можно выделить следующие типы нагрузочного тестирования, классифицированные по целям проведения тестирования:

А. Тестирование на соответствие требованиям

Целью данного вида тестирования является проверка информационной системы на соответствие SLA (Service Level Agreement) — соглашения об уровне сервиса. Эти соглашения обычно включают в себя некоторые количественные показатели (метрики), отражающие уровень сервиса. Тестирование на соответствие требованиям особенно актуально в случаях сдачи системы в промышленную эксплуатацию и при приемосдаточных испытаниях.

Модель нагрузки при тестировании на соответствие требованиям строится на основе спецификаций, описанных в SLA. Например, если в качестве SLO выбрано значение выражающее количество работающих пользователей, а SLA предполагают, что система

должна выдерживать нагрузку, возникающую при одновременной работе N пользователей, то модель нагрузки должна включать сценарии, имитирующие работу именно такого количества пользователей. При этом число пользователей может динамически изменяться.

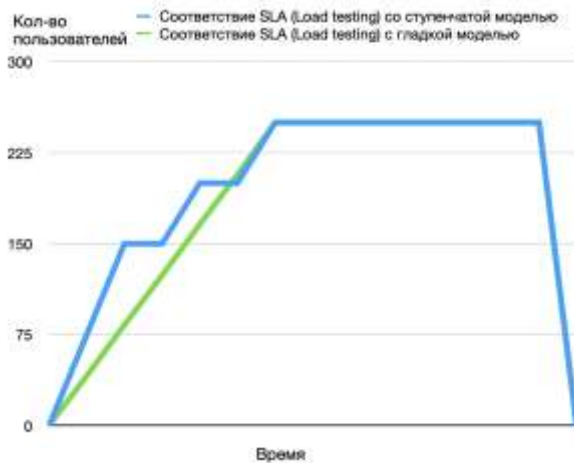


Рис. 1. Модели нагрузки тестирования на соответствие требованиям

Важно учитывать, что модель нагрузки, основанная исключительно на SLA, может не полностью соответствовать реальным условиям эксплуатации ИС. Например, в SLA может быть указана нагрузка в N пользователей, но в реальности система может столкнуться с пиковыми нагрузками, превышающие это значение.

Одной из ключевых проблем при составлении профиля нагрузки для тестирования на соответствие требованиям является его зависимость исключительно от SLA. Это может привести к отсутствию проверки реальных сценариев использования системы. Кроме того, в SLA не всегда присутствуют дополнительные метрики, которые в ряде случаев важны для оценки производительности системы.

В. Тестирование на поиск узких мест (Bottleneck Testing)

Этот вид нагрузочного тестирования направлен на выявление и анализ компонентов системы, которые ограничивают ее производительность. Узкие места определяются как потенциальные источники падения производительности. Тестирование на поиск узких мест направлено на локализацию проблемных участков ИС, например, таких как запросы к базе данных, перегруженные сетевые интерфейсы или неэффективные алгоритмы обработки данных. Тестирование не сводится только к выявлению проблемного компонента; его задачами являются также количественная оценка влияния узкого места на производительность и поиск возможностей оптимизации.

Модель нагрузки при тестировании на поиск узких мест формируется так, чтобы нагрузить именно тот компонент, который потенциально может быть узким местом. При этом важно учитывать, что такая избирательность тестирования может привести к упрощению модели нагрузки, что не всегда соответствует реальным условиям эксплуатации. В

связи с этим результаты тестирования на поиск узких мест не всегда применимы для комплексной оценки производительности ИС.

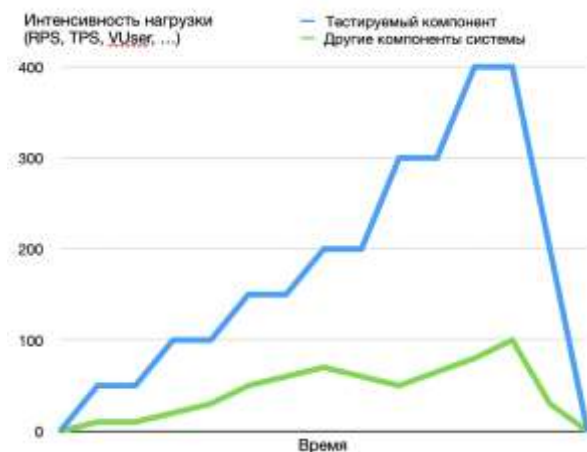


Рис. 2. Модель нагрузки тестирования на поиск узких мест

Составление профиля нагрузки для тестирования поиска узких мест связано с необходимостью разработки уникальных сценариев нагрузки, которые отличаются от типовых. Например, при выборе сценариев нагрузки стоит отдать предпочтение тем, которые используют наименьшее число интеграционных вызовов, что снизит влияние смежных компонент на результаты тестирования, в случае отсутствия возможности отказа от интеграций, рекомендуется использовать заглушки, имитирующие поведение интеграционной системы. Также во время проведения данного вида теста стоит дополнительно проводить мониторинг работы потенциально-проблемного компонента ИС, чтобы постараться определить причину возникновения узкого места.

С. Тестирование максимальной производительности (Peak Performance Testing)

Основная цель данного вида тестирования — определение точки максимальной производительности системы, выявление возможных точек отказа и прогнозирование зависимости деградации системы от величины нагрузки. Данный вид тестирования может быть проведён при выпуске ИС в промышленную эксплуатацию, поскольку отражает пиковые возможности системы. Это также может оказаться важным при планировании масштабирования ИС. Кроме того, данный вид тестирования показывает, как система будет вести себя в экстремальных условиях, когда превышаются обычные эксплуатационные параметры.

Одной из задач при формировании сценариев тестирования максимальной производительности является определение критериев успешности. Как правило критериями успешности выступают характеристики, взятые из SLA.

Модель нагрузки при тестировании максимальной производительности обычно имеет ступенчатый вид (рис. 3). Это означает, что нагрузка увеличивается постепенно, с определенным шагом и растет до момента, когда наступает резкое несоответствие поведения ИС критериям успешности теста.

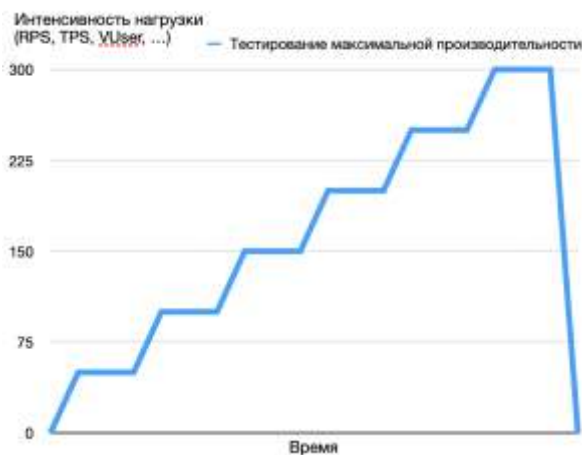


Рис. 3. Модель нагрузки тестирования максимальной производительности

Если система находится в стадии промышленной эксплуатации, то рекомендуется ориентироваться на стандартный профиль нагрузки.

D. Тестирование на больших объемах (Volume Testing)

Этот вид нагрузочного тестирования, направленный на исследование поведения системы при увеличении объема данных, с которыми она работает.

Основная цель данного вида тестирования заключается в оценке того, как система справляется с обработкой больших объемов данных, и выявлении возможного падения производительности, связанного с увеличением объема данных, с которыми она работает. Данный вид тестирования особенно важен для систем, которые, во-первых, активно взаимодействуют с базами данных (БД) и, во-вторых, обрабатывают значительные объемы данных в реальном времени.

Проводить тестирование объемов стоит для исследования следующих аспектов работы ИС:

- 1) Влияние объемов данных на быстродействие. Тестирование позволяет определить зависимость времени выполнения программной транзакции от объема обрабатываемых данных;
- 2) Необходимый уровень масштабируемости. Тестирование позволит определить характер поведения ИС при увеличении объема данных, например, при росте числа пользователей или накоплении данных в процессе эксплуатации;
- 3) Оценка конфигурации сервера БД. При тестировании на больших объемах зачастую узким местом оказывается БД. Тестирование помогает локализовать потенциальную проблему, связанную с БД.

Дополнительно во время проведения тестирования на больших объемах рекомендуется проводить мониторинг метрик на компонентах ИС, работающие с данными. Например, время выполнения запросов к БД, скорость чтения/записи данных с дисков на сервере БД, блокировки данных в СУБД.

С точки зрения организации проведения тестирования предлагается два подхода.

Первый предусматривает генерацию данных во время тестирования. В ходе выполнения теста объем данных в БД увеличивается, и одновременно выполняются

запросы к БД. Это позволяет анализировать, как производительность системы изменяется по мере роста объема данных.

Второй основан на предварительном заполнении БД различными данными, либо на подготовке файлов различных объемов, подаваемых на вход ИС. При таком подходе более наглядно получается сравнить время выполнения транзакции на разных объемах данных.

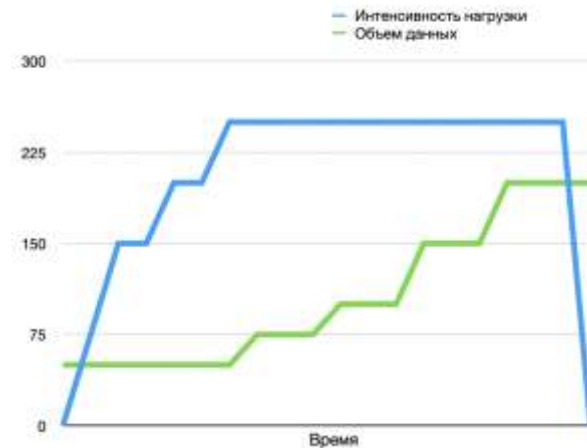


Рис. 4. Модель нагрузки тестирования на больших объемах

При составлении профиля нагрузки тестирования на больших объемах следует:

- 1) Сделать упор на сценариях, ориентированных на работу с данными;
- 2) В случае если ИС производит обработку данных и на сервере БД и на сервере приложения, необходимо форматировать сценарии отдельно для каждого;
- 3) Рекомендуется проводить точечное тестирование, для исключения влияния узких мест системы, например, таких как скорость чтения данных, на оценку производительности со стороны СУБД.

E. Моделирование и прогнозирование производительности

Этот вид нагрузочного тестирования направлен на определение параметров масштабируемости ИС. Этот подход позволяет предположить уровень производительности ИС в различных условиях эксплуатации.

Данный вид тестирования следует применять для задач сравнения производительности ИС при горизонтальном масштабировании, определения оптимальной конфигурации серверов ИС.

Модель нагрузки при моделировании и прогнозировании производительности строится на основе стандартных сценариев, которые максимально приближены к реальным условиям эксплуатации. При этом важно учитывать, что интенсивность нагрузки должно соответствовать конфигурации серверов ИС. Например, если стенд X может обрабатывать только 50% от ожидаемой промышленной нагрузки, коэффициент интенсивности может быть установлен на уровне 0.5.

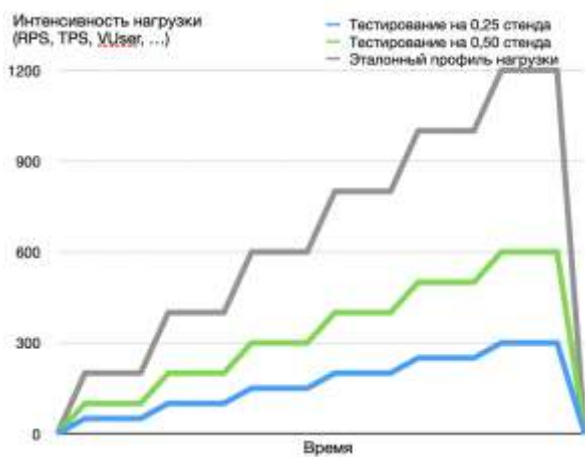


Рис. 5. Модель нагрузки тестирования моделирования и прогнозирования производительности

Основные рекомендации при моделировании и прогнозировании производительности сводятся к следующему:

- 1) Следует провести несколько итераций тестирования на разных конфигурациях для более точного определения параметров масштабируемости системы;
- 2) При определении параметров масштабируемости необходимо учитывать, что производительность системы может изменяться нелинейно при увеличении нагрузки или ресурсов.

IV. ЗАКЛЮЧЕНИЕ

В статье проанализированы существующие методы нагрузочного тестирования информационных систем. Отмечено, что НТ является ключевым способом тестирования информационных систем, обеспечивающим как соответствие бизнес-требованиям, так и требованиям производительности.

Предложена классификация НТ, включающая следующие пять основных видов:

- 1) Тестирование на соответствие требованиям
- 2) Тестирование на поиск узких мест
- 3) Тестирование максимальной производительности
- 4) Тестирование на больших объемах
- 5) Моделирование и прогнозирование производительности

Показано, что выбор конкретного вида (или комбинации видов) нагрузочного тестирования должен основываться на четко сформулированных задачах, стоящих перед командой на текущем этапе жизненного цикла ИС. Успешное проведение любого вида НТ требует тщательной проработки профиля нагрузки, выбора релевантных метрик, обеспечения изолированности и контролируемости тестового окружения ИС, а также корректной интерпретации результатов с учетом специфики выбранного вида тестирования.

V. БИБЛИОГРАФИЯ

- [1] Molyneaux, I. The Art of Application Performance Testing: From Strategy to Tools. O'Reilly Media, 2014.
- [2] Jain, R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley, 1991.
- [3] Sharma, N., Marin, B., et al. Evaluating User Behavior and Performance in Cloud Applications // IEEE Transactions on Cloud Computing, 2021. Vol. 9, No. 2. P. 756-769.
- [4] Mayer, B., Weinreich, R. An Approach to Microservice Architecture Performance Testing // Proceedings of the 12th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2017. P. 171-177.
- [5] Menascé, D. A., Almeida, V. A. F., Dowdy, L. W. Capacity Planning and Performance Modeling: From Mainframes to Client-Server Systems. Prentice Hall, 1994.
- [6] Mansouri, Y., Prokhorenko, V., Babar, M. A. An automated implementation of hybrid cloud for performance evaluation of distributed databases // Future Generation Computer Systems. 2020. Vol. 113. P. 492-506.
- [7] Katevas, V., Fatouros, G., Kyriazis, D., Kousiouris, G. Embedding automated function performance benchmarking, profiling and resource usage categorization in function as a service DevOps pipelines // Future Generation Computer Systems. 2024. Vol. 160. P. 223-237.
- [8] Dhulavagol, P. M., Bhajantri, V. H. Performance Analysis of Distributed Processing System using Shard Selection Techniques on ElasticSearch // Journal of Parallel and Distributed Computing, 2023. Vol. 176. P. 104-118.
- [9] Sergievsky, M. V. Modeling Unified Language Templates for Designing Information Systems / M. V. Sergievsky // Automatic Documentation and Mathematical Linguistics. – 2020. – Vol. 54, No. 1. – P. 26-35.
- [10] Bao, L., Liu, Y., et al. Performance Anomaly Detection and Prediction for Cloud Systems: A Machine Learning Approach // IEEE Transactions on Services Computing. 2022. Vol. 15, No. 3. P. 1233-1245.
- [11] Molnar, D., & Liptak, T. A Survey of Performance Testing Strategies for Microservices // Journal of Systems and Software. 2023. Vol. 195. Article 111547.
- [12] Sefraoui, O., Aalaoui, M. E. Leveraging AI for Dynamic Load Profile Generation in Performance Testing // Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIIC). 2024. P. 412-417.
- [13] Святенко, А. С. Современные методы и инструменты нагрузочного тестирования программного обеспечения / А. С. Святенко // Перспективы науки. – 2020. – № 9(132). – С. 25-27. – EDN DMCSUD.
- [14] Rostami, A., Loiret, F., et al. Performance Testing in CI/CD Pipelines: A Systematic Mapping Study // ACM Computing Surveys. 2023. Vol. 55, No. 10. Article 215.

Load testing of information systems: types and methods

N.I. Kakurin, N.A. Karpov, D.D. Rudenko

Abstract—The article is devoted to the analysis of types of load testing of information systems. A classification of load testing types is presented. An overview of each type is provided, along with descriptions of possible application areas and recommendations for selecting load models and additional metrics. Potential challenges encountered in forming the load profile are also identified.

The material will be useful for specialists in the field of information system testing, as well as for engineers whose task is to achieve maximum efficiency of information systems.

Keywords— load testing of information systems, performance testing, classification of load testing types, load profile.

REFERENCES

- [1] Molyneaux, I. The Art of Application Performance Testing: From Strategy to Tools. O'Reilly Media, 2014.
- [2] Jain, R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley, 1991.
- [3] Sharma, N., Marin, B., et al. Evaluating User Behavior and Performance in Cloud Applications // IEEE Transactions on Cloud Computing, 2021. Vol. 9, No. 2. P. 756-769.
- [4] Mayer, B., Weinreich, R. An Approach to Microservice Architecture Performance Testing // Proceedings of the 12th IEEE/ACM International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS), 2017. P. 171-177.
- [5] Menascé, D. A., Almeida, V. A. F., Dowdy, L. W. Capacity Planning and Performance Modeling: From Mainframes to Client-Server Systems. Prentice Hall, 1994.
- [6] Mansouri, Y., Prokhorenko, V., Babar, M. A. An automated implementation of hybrid cloud for performance evaluation of distributed databases // Future Generation Computer Systems. 2020. Vol. 113. P. 492–506.
- [7] Katevas, V., Fatouros, G., Kyriazis, D., Kousiouris, G. Embedding automated function performance benchmarking, profiling and resource usage categorization in function as a service DevOps pipelines // Future Generation Computer Systems. 2024. Vol. 160. P. 223-237.
- [8] Dhulavvagol, P. M., Bhajantri, V. H. Performance Analysis of Distributed Processing System using Shard Selection Techniques on ElasticSearch // Journal of Parallel and Distributed Computing. 2023. Vol. 176. P. 104–118.
- [9] Sergievsky, M. V. Modeling Unified Language Templates for Designing Information Systems / M. V. Sergievsky // Automatic Documentation and Mathematical Linguistics. – 2020. – Vol. 54, No. 1. – P. 26-35.
- [10] Bao, L., Liu, Y., et al. Performance Anomaly Detection and Prediction for Cloud Systems: A Machine Learning Approach // IEEE Transactions on Services Computing. 2022. Vol. 15, No. 3. P. 1233-1245.
- [11] Molnar, D., & Liptak, T. A Survey of Performance Testing Strategies for Microservices // Journal of Systems and Software. 2023. Vol. 195. Article 111547.
- [12] Sefraoui, O., Aalaoui, M. E. Leveraging AI for Dynamic Load Profile Generation in Performance Testing // Proceedings of the International Conference on Artificial Intelligence in Information and Communication (ICAIIIC). 2024. P. 412-417.
- [13] Svyatenko, A. S. Modern methods and tools of software load testing / A. S. Svyatenko // Perspectives of Science. – 2020. – No. 9(132). – pp. 25–27. – EDN DMCSUD.
- [14] Rostami, A., Loiret, F., et al. Performance Testing in CI/CD Pipelines: A Systematic Mapping Study // ACM Computing Surveys. 2023. Vol. 55, No. 10. Article 215.