

Электронный научный журнал «Век качества» ISSN 2500-1841 <https://www.agequal.ru>

2025, №4 https://www.agequal.ru/pdf/2025/AGE_QUALITY_4_2025.pdf

Ссылка для цитирования этой статьи:

Ванина М.Ф., Ерохин А.Г., Фролова Е.А. Комплексный подход к конкурентному анализу на рынке SaaS в условиях ограниченности данных: интеграция цифрового следа и операционных метрик // Электронный научный журнал «Век качества». 2025. №4. С. 329-350. Режим доступа: <https://www.agequal.ru/pdf/2025/425018.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

УДК 004.7

**Нагрузочное тестирование как способ оценки качества
программного обеспечения распределенных информационных систем
в условиях импортозамещения**

Ванина Маргарита Федоровна,
доцент, кандидат технических наук,
доцент кафедры «Бизнес-информатика»,
Московский технический университет связи и информатики,
111024, Россия, г. Москва, ул. Авиамоторная, д. 8а
m.f.vanina@mtuci.ru

Ерохин Андрей Густавович,
доцент, кандидат технических наук,
доцент кафедры «Бизнес-информатика»,
Московский технический университет связи и информатики,
111024, Россия, г. Москва, ул. Авиамоторная, д. 8а
a.g.erokhin@mtuci.ru

Фролова Елена Александровна,
старший преподаватель кафедры «Бизнес-информатика»,
Московский технический университет связи и информатики,
111024, Россия, г. Москва, ул. Авиамоторная, д. 8а
e.a.frolova@mtuci.ru

IT-отрасль в России до сих пор остается одной из самых импортозависимых отраслей экономики РФ. Именно поэтому задача импортозамещения здесь стоит особенно остро и к настоящему времени всё большее число компаний различной отраслевой направленности переходит на отечественное программное обеспечение. Однако важно не просто разработать информационную систему на отечественных решениях, но и обеспечить миграцию данных в такую систему без остановки бизнес-процессов так, чтобы новая система справлялась со всеми нагрузками. Иными словами, необходимо обеспечить соответствующее качество отечественных решений, чтобы по

совокупности показателей они не уступали или даже превосходили зарубежные. Именно это призвано обеспечить процесс тестирования информационных систем при различных нагрузках. Кроме этого, в современном мире все более важным становится смена подхода к построению информационных систем: от традиционных, статичных и централизованных моделей к децентрализованным архитектурам. Основой этих новых систем являются методы сбора и анализа данных, что приводит к формированию распределенных информационных систем (РИС). В статье проводится анализ современных подходов к процессам нагрузочного тестирования в условиях перехода на отечественное программное обеспечение и построения распределенных информационных систем на его основе.

Ключевые слова: импортозамещение, программное обеспечение, распределенные информационные системы, нагрузочное тестирование, метрики качества, санкционные риски, цифровой суверенитет.

Введение

Активное развитие собственной IT-отрасли в России является критически важным для становления её суверенитета во всех областях. Многие зарубежные разработчики закрыли нам доступ к своему программному обеспечению (ПО) и нет никакой гарантии, что так не поступят и те, кто пока остается на российском рынке. К тому же использование зарубежного софта в критически важных отраслях экономики является опасным, поскольку помимо санкционных рисков несет и риски несанкционированного доступа к конфиденциальной и секретной информации. Именно поэтому в России поставлена и активно решается задача импортозамещения в области информационных технологий, предусматривающая, в том числе, разработку отечественного программного обеспечения [1].

Однако недостаточно только разработать соответствующий софт. Необходимо, чтобы отечественные программы не просто заменяли зарубежные, но не уступали бы им по функционалу, удобству работы, быстродействию, надежности, достоверности результатной информации, т.е. были бы не хуже и даже превосходили бы зарубежные решения по всей совокупности показателей качества [2; 3]. Это может быть достигнуто путем нагрузочного тестирования соответствующего ПО. Под нагрузочным тестированием понимается процесс

оценки производительности программного обеспечения или системы под определенными нагрузками для определения её поведения в условиях интенсивного использования [4].

Оценка эффективности функционирования комплексных информационных систем традиционно представляет собой сложную задачу [5; 6]. Ключевым моментом является четкое понимание того, что именно подлежит измерению, каким образом это следует делать, как интегрировать разнородные и разнокачественные параметры и, самое главное, – с какой целью проводится оценка. Данные вопросы в полной мере актуальны как для систем управления базами данных (СУБД), так и для других транзакционных систем, таких как серверы приложений (СП) и кластеры серверов приложений (КСП).

С момента появления подобных систем возникла потребность в разработке инструментов для оценки их производительности, предлагающих разнообразные методики измерений. Это привело к появлению количественных показателей, характеризующих быстродействие СУБД и транзакционных систем. Одновременно с созданием тестов возникла необходимость в определении требований к ним, чтобы гарантировать практическую ценность получаемых результатов [5; 6].

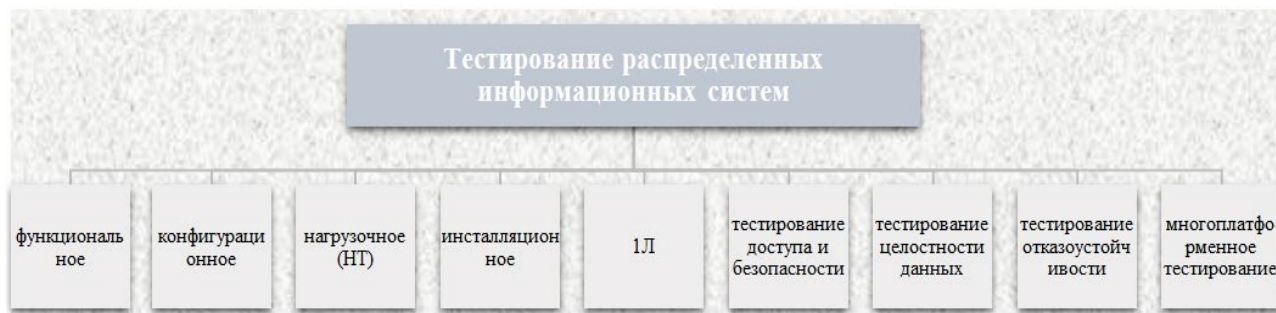
В условиях незаконного санкционного давления и активного перехода к отечественному ПО [1] нагрузочное тестирование приобретает особое значение. Ведь важно не просто разработать информационную систему на отечественных решениях, но и обеспечить миграцию данных в такую систему без остановки бизнес-процессов таким образом, чтобы новая система справлялась со всеми нагрузками. Примером такого перехода является проект Московского метрополитена [7] по переходу на отечественную операционную систему ASTRA Linux и СУБД Postgres Pro. В техническом задании на данную работу прямо указано, что ключевым требованием является совместимость поставляемого программного обеспечения с уже установленным оборудованием и программными комплексами.

В связи с этим возникает другая проблема – оценка качества разрабатываемого или приобретаемого импортозамещенного ПО. Рассмотрим основные аспекты данной проблемы.

Категории нагрузочного тестирования

В современном мире все более важным становится смена подхода к построению информационных систем (ИС): от традиционных, статичных и централизованных моделей к децентрализованным архитектурам. Основой этих новых систем являются методы сбора и анализа данных, что приводит к формированию распределенных информационных систем (РИС). Спектр использования РИС постоянно расширяется. На базе технологий РИС построены современные платформы облачных вычислений [8; 9], социальные сети, финансовые сети, блокчейн-системы [10].

При разработке тестовых заданий для подобных систем необходимо находить баланс между общими принципами и спецификой конкретной области применения. На рис. 1 представлены наиболее востребованные категории тестов для РИС в настоящее время [11; 12].



Источник: составлено авторами на основе данных [11; 12]

Рис. 1. Наиболее востребованные категории тестов для РИС

В настоящий момент разработано достаточно большое количество подходов к реализации процесса нагрузочного тестирования. Одним из наиболее часто используемых подходов является тест TRC (Transaction Processing Performance Council) [12], позволяющий измерять

производительность не только процессора, но и всей вычислительной системы. Такие тесты используются для оценки эффективности функционирования систем в различных областях компьютерной коммерческой деятельности, например, в обработке данных и транзакций, поддержке принятия решений, вычислений в различных предметных областях.

Существует большое число тестов TRC [12], например, TRC-B для проверки возможностей различных СУБД, TRC-C для моделирования различных бизнес-процессов (банковских операций, бухгалтерских процессов и т.д.), TRC-W для анализа систем электронной коммерции. Отличительной чертой тестов TRC является их специализированный характер, поэтому они подходят только для распределенных информационных систем, в основу которых положены стандартные протоколы обмена информацией. С использованием TRC-тестов можно провести сравнительный анализ производительности различных аппаратно-программных платформ по совокупности унифицированных показателей. Если же речь идет об оценке производительности конкретной РИС, то применение такого подхода к тестированию становится невозможным.

Если провести стоимостный анализ процессов TRC-тестирования, то оказывается, что себестоимость подобного рода услуг является достаточно высокой. Кроме этого, сам процесс подобного тестирования представляет собой сложную высокоуровневую инженерно-техническую задачу. Поэтому в настоящее время появляется множество компаний, предлагающих услуги по тестированию аппаратно-программной конфигурации распределенных информационных систем и разрабатывающих соответствующий софт для этого. Тестирование по одному только тесту может обойтись компании в десятки миллионов рублей. В частности, уже упоминавшийся проект [7] предусматривает выделение 364 млн руб., значительная часть из которых пойдет на тестирование приобретаемого софта.

Нагрузочное тестирование – это метод оценки работоспособности РИС в условиях повышенной нагрузки. Его механизм прост: запускается одновременно несколько клиентских приложений (КП РИС), что создаёт нагрузку, близкую к реальной эксплуатации системы.

В ходе тестирования фиксируются ключевые метрики, позволяющие проанализировать:

- динамику отклика системы при нарастающей нагрузке;
- стабильность работы в условиях многопользовательского доступа;
- соответствие фактическим требованиям технического задания [13].

Такой подход даёт целостное представление о качестве РИС:

- выявляет статистические закономерности в её поведении;
- оценивает интегральные показатели эффективности;
- проверяет выполнение спецификаций, заданных в документации [14; 15].

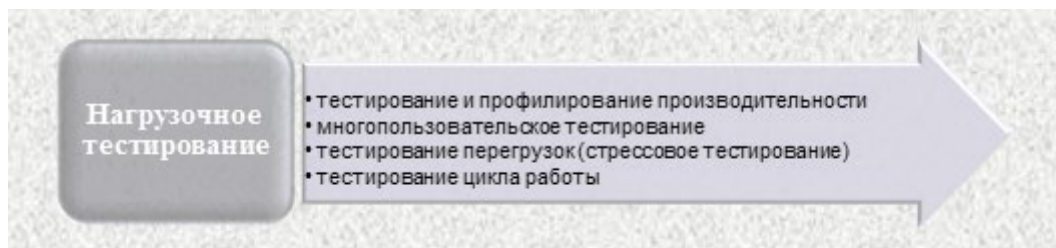
Практические выгоды метода многогранны:

- оптимизация аппаратной конфигурации системы;
- диагностика и локализация скрытых дефектов;
- подтверждение способности РИС обслуживать все категории пользователей при пиковых нагрузках.

Однако реализация тестирования сопряжена с рядом сложностей:

- необходимостью проектирования разнообразных тестовых моделей (ТМ) и сценариев (ТВ);
- потребностью в специализированном оборудовании для имитации реальных условий эксплуатации.

Нагрузочное тестирование включает в себя четыре основных процесса (рис. 2):



Источник: составлено авторами

Рис. 2. Виды нагрузочного тестирования

При использовании тестов производительности проводится проверка соответствия проверяемой системы требованиям к показателям производительности. Процесс многопользовательского тестирования предполагает наличие реальных или создание виртуальных пользователей тестируемой РИС.

В ходе стрессового тестирования выполняется функциональный анализ распределённой информационной системы в условиях максимальных нагрузок. Основная задача таких испытаний состоит в определении технических границ системы с целью убедиться в корректности её работы при экстремальных сценариях.

Подобные проверки особенно актуальны, когда планируется эксплуатация РИС в среде с высокой пользовательской активностью. В процессе тестирования в систему последовательно внедряют значительные объёмы данных.

Существует два ключевых направления подобных испытаний:

- 1) многопользовательское тестирование;
- 2) перегрузочное тестирование.

Они позволяют комплексно оценить:

- пропускную способность РИС;
- потенциал масштабирования системы.

Для проведения тестов задействуют большое количество участников – как реальных, так и виртуальных. При этом часть пользователей генерирует запросы с параметрами, выходящими за стандартные рамки.

Перегрузочное тестирование характеризуется обработкой колоссальных массивов информации и выполнением множества функциональных запросов в сжатые временные интервалы.

Отдельный вид проверки – тестирование рабочего цикла. Оно проводится при строго заданных условиях и направлено на анализ следующих показателей:

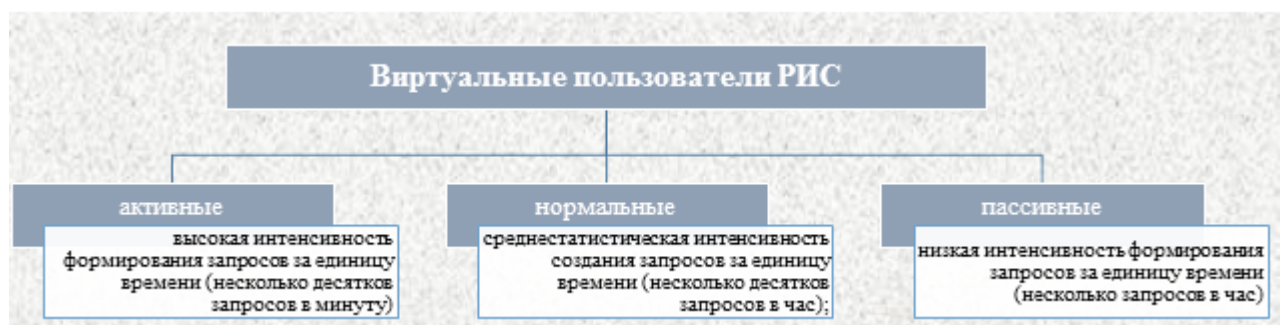
- устойчивость информационно-вычислительной сети;
- временные задержки при передаче данных и другие параметры.

При организации нагрузочного тестирования разрабатывают специальные модели нагрузки. Как правило, они включают в себя набор сценариев для виртуальных пользователей.

Среди виртуальных пользователей принято выделять три категории:

- 1) активные (с максимальной интенсивностью запросов);
- 2) нормальные (со средней интенсивностью запросов);
- 3) пассивные (с минимальной интенсивностью запросов).

Разделение основано на различиях в частоте и объёме генерируемых запросов к системе (рис. 3).



Источник: составлено авторами

Рис. 3. Характеристика виртуальных пользователей РИС

При использовании нагрузочного тестирования проверяются такие показатели качества функционирования РИС, как характеристики

использования ресурсов, стабильность работы и отказоустойчивость системы. Эти показатели оказывают существенное влияние на фактическую работоспособность потенциально безупречной информационной системы. При этом необходимо четко сформулировать критерии оценки таких требований. В частности, требование к отказоустойчивости может быть сформулировано двояко. Оно может формулироваться как в общем виде (в том смысле, что система должна работать в любой ситуации), так и содержать специфичные формулировки, например, требование отследить поведение системы при пропаже соединения с сервером базы данных [16], или другие сценарии в процессе работы. Для этого составляется список самых распространенных сценариев работы, возникающих в процессе функционирования реальной РИС. После отработки этих сценариев в случае, если они отработаны успешно и нефункциональные требования к РИС соответствуют имеющимся требованиям (причем с определенным запасом), приступают к отработке сценариев более сложного характера, которые не относятся к категории самых распространенных.

Сравнительный анализ подходов к нагрузочному тестированию

Все подходы к рабочему нагрузочному тестированию любой РИС используют GUI КП РИС. При этом можно выделить два типа подходов к подобного рода тестированию (А и В):

- тип А подразумевает привлечение к процессу тестирования особых групп сотрудников (тестировщиков) и разработку и применение специальных тестовых вариантов и моделей, имитирующих работу реальных пользователей системы;
- тип В подразумевает привлечение к процессу тестирования реальных пользователей РИС (так называемых демо-пользователей). Данное тестирование целесообразно проводить на стадии внедрения распределенной информационной системы.

В таблице приведен сравнительный анализ подходов к процессу рабочего нагрузочного тестирования.

На основе обобщения данных таблицы можно прийти к следующим выводам:

1. Стоимость процесса тестирования при втором подходе значительно выше (иногда на несколько порядков) при увеличении привлеченных к тестированию демо-пользователей.
2. Первый подход обеспечивает большую степень управляемости процесса тестирования РИС.
3. Подготовка к процессу тестирования при первом подходе требует меньшего количества времени, хотя и при втором подходе оно также не слишком велико.
4. При необходимости получения экспериментальных данных о работе всего спектра клиентских приложений РИС необходимо использовать второй подход.

Сравнительный анализ подходов к процессу рабочего нагрузочного тестирования

Подход	Недостатки	Достоинства
Тип А	<ol style="list-style-type: none">1. Значительный рост себестоимости тестирования при увеличении числа пользователей РИС.2. Необходимость разработки специальных процедур, использования дополнительного аппаратного и программного обеспечения с соответствующим увеличением затрат или даже модернизация РИС для получения экспериментальных данных о функционировании всего спектра клиентских приложений (КП РИС).3. Необходимость значительного числа компьютеров для проведения тестирования и тестируемых, что также ведет к увеличению затрат.	<ol style="list-style-type: none">1. Возможность поддержки различных тестовых моделей и моделей виртуальных пользователей РИС.2. Возможность управления процессами работы пользователей с РИС.3. Низкая себестоимость процесса тестирования при небольшом количестве пользователей (как правило, не более 10).4. Не требует значительного времени для подготовки к процессу тестирования.5. Все тестовые эксперименты возможно воспроизвести и повторить в любой момент.
Тип В	<ol style="list-style-type: none">1. Управление работой пользователей РИС является достаточно сложным.2. Отсутствие возможности использования моделей взаимодействия виртуальных пользователей РИС.3. Процедура получения данных о функционировании клиентских приложений (КП) является очень сложной. Поэтому цели тестирования не всегда могут быть достигнуты.4. Проведение тестирования на ранних стадиях разработки РИС невозможно из-за обязательного требования к наличию полностью отлаженной РИС со всем спектром инсталлируемых КП РИС.5. Требуется поддержка дополнительных серверов РИС.6. Требуется учитывать возможность наличия различных версий клиентских приложений (КП), что, в свою очередь, влечет необходимость разработки механизмов синхронизации КП.7. Сложность прогнозирования результатов тестирования.8. Сложность разработки плана тестирования.9. Повторение тестовых экспериментов принципиально невозможно, поскольку невозможен возврат РИС к текущему состоянию.10. Необходимость учитывать в процессе тестирования побочные действия: возможные несанкционированные воздействия, попытки обхода системы безопасности, возможные сбои в сети и др.	<ol style="list-style-type: none">1. При увеличении количества пользователей РИС затраты на тестирования возрастают незначительно.2. Подготовка к процессу тестирования не требует существенных временных затрат.3. При использовании внешних ресурсов снижаются требования к числу тестируемых и тестового оборудования.

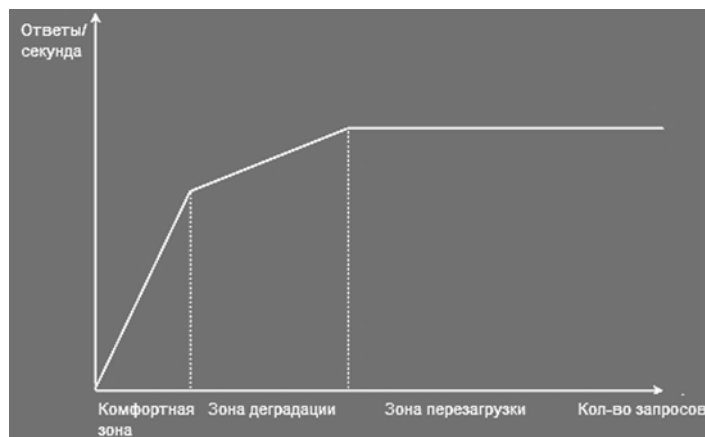
Источник: составлено авторами

Оценка качества используемого ПО

Основной целью проведения нагрузочного тестирования является оценка количественных характеристик исследуемой системы. Однако после этого необходимо перейти к качественному осмыслению результатов тестов. Для тестов с различными моделями нагрузки зависимости основных параметров от количества запросов и от времени имеют различные функциональные зависимости. Для удобства изобразим тип зависимостей на графиках.

Возможны два вида моделей тестирования: с закрытой и открытой нагрузкой. Использование модели с закрытой нагрузкой предполагает ситуацию, когда максимальное количество виртуальных пользователей определяется заранее и на протяжении всего цикла тестирования остается постоянной величиной. Применение моделей с открытой нагрузкой подразумевает создание новых виртуальных пользователей вне зависимости от имеющегося числа пользователей, одновременно использующих систему. Вторая модель больше соответствует реальности, однако она сложнее, поскольку в ней отсутствует контроль над числом пользователей системы вообще.

Рассмотрим результаты для моделей с закрытой нагрузкой. На основе анализа результатов работ [17; 18] и проведения ряда исследований можно получить следующие результаты. На рис. 4 приведена зависимость скорости ответа сервиса от количества запросов.

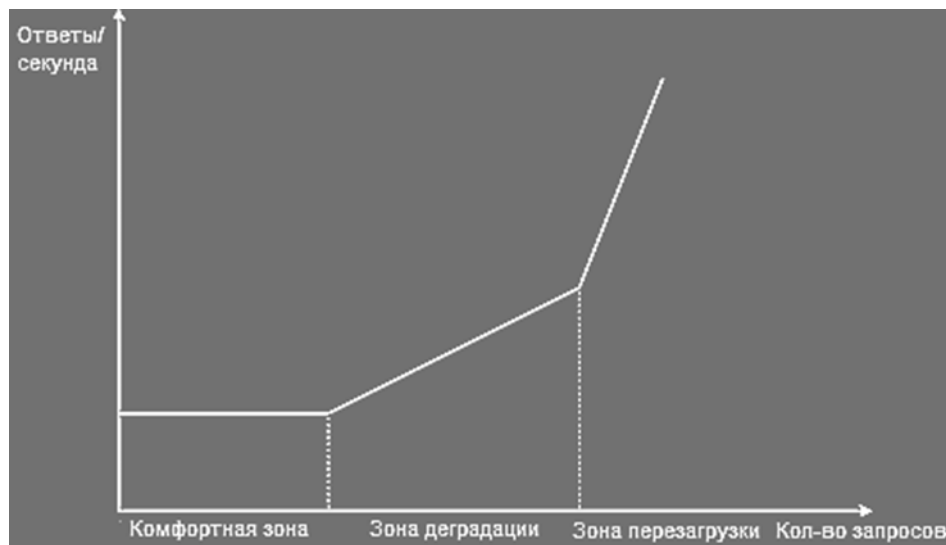


Источник: составлено авторами на основе данных [17; 18]

Рис. 4. Зависимость скорости ответа сервиса от количества запросов при использовании модели тестирования с закрытой нагрузкой

Во всех тестах можно было выделить две четкие границы. Изначально при увеличении количества запросов коэффициент ответа растет линейно до достижения некоторого порогового значения. После этого рост продолжается, но производная падает. Параллельно с этим могут появляться ошибки. Связано это с тем, что не все отдельные потоки в рамках одной виртуальной машины могут успешно завершиться, так как в традиционной модели памяти java используется вытесняющая многопоточность. При дальнейшем увеличении нагрузки скорость перестает расти и переходит в режим насыщения, когда каждый новый пользователь просто ставится в очередь системой и ожидает момента освобождения одного из потоков.

Несколько иным образом ведет себя такой параметр, как среднее время ответа на запрос (рис. 5). Изначально, как показывает тестирование, оно постоянно, с подключением новых пользователей оно начинает линейно возрастать, так как для вновь прибывающих запросов новых потоков не создается и пользователи становятся в очередь. При достижении области перегрузки время ответа начинает резко возрастать, при этом нарастает количество ошибок. Это связано с переполнением очередей приложения. Лишние запросы просто теряются, выходя за пределы очереди.

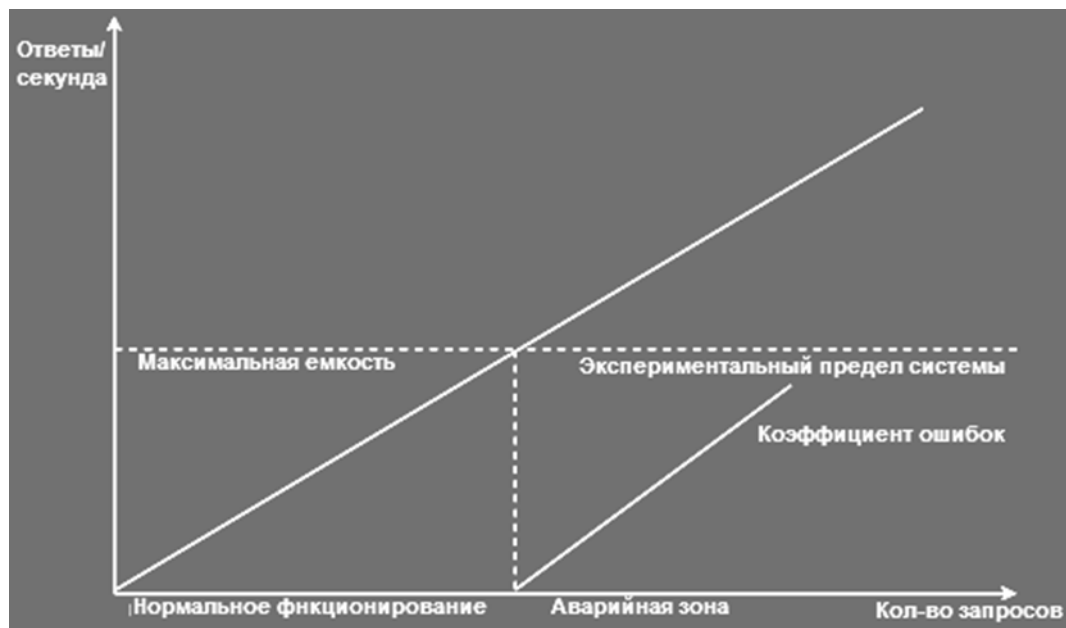


Источник: составлено авторами на основе данных [17; 18]

Рис. 5. Зависимость скорости ответа на запрос от количества запросов при использовании модели тестирования с закрытой нагрузкой

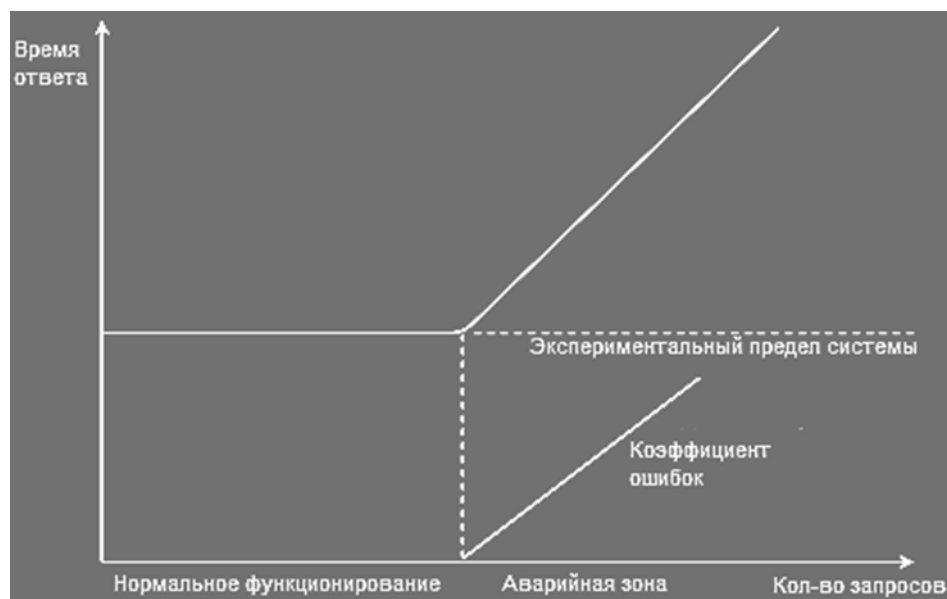
В случае открытой модели нагрузки при увеличении количества запросов количество ответов продолжает увеличиваться линейно. Но при этом наблюдается интересное явление — с некоторого значения нагрузки начинают появляться ошибки, и их количество растет линейно. Это объясняется отсутствием свободных обработчиков, способных подхватить запрос, из-за этого такие запросы отбраковываются еще на уровне балансировщика нагрузки или на уровне сервера приложений (WebLogic). Данная зависимость наиболее полезна в практическом смысле, т.к. позволяет определить границы штатных режимов работы системы. При промышленной эксплуатации системы выход в аварийную зону должен быть запрещен на уровне настроек балансировщиков нагрузки.

Результаты аналогичных экспериментов для модели с открытой нагрузкой приведены соответственно на рис. 6 и 7.



Источник: составлено авторами на основе данных [17; 18]

Рис. 6. Зависимость скорости ответа сервиса от количества запросов при использовании модели тестирования с открытой нагрузкой

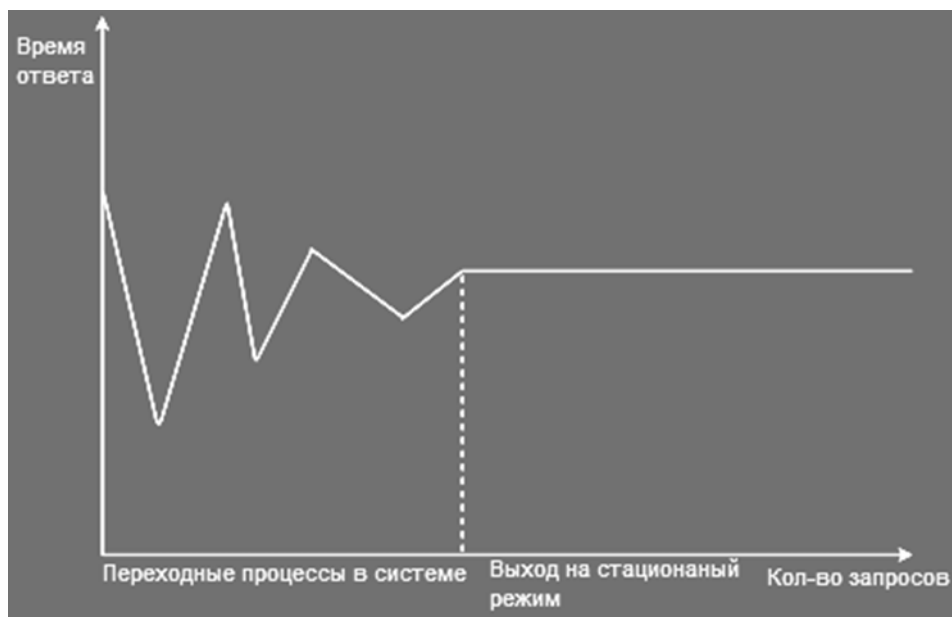


Источник: составлено авторами на основе данных [17; 18]

Рис. 7. Зависимость скорости ответа на запрос от количества запросов при использовании модели тестирования с открытой нагрузкой

Определившись с типами нагрузки и их поведением при разных функциональных параметрах, можно перейти к интерпретации полученных результатов в терминах стационарных и нестационарных процессов.

Качественный анализ можно проводить по таким параметрам, как среднее время ответа на запрос, скорость ответа на запрос, количество используемой оперативной памяти и среднее время ответа системы в зависимости от общего времени её работы. Сначала рассмотрим один из самых важных параметров системы – среднее время ответа на запрос. Качественная зависимость изображена на рис. 8.

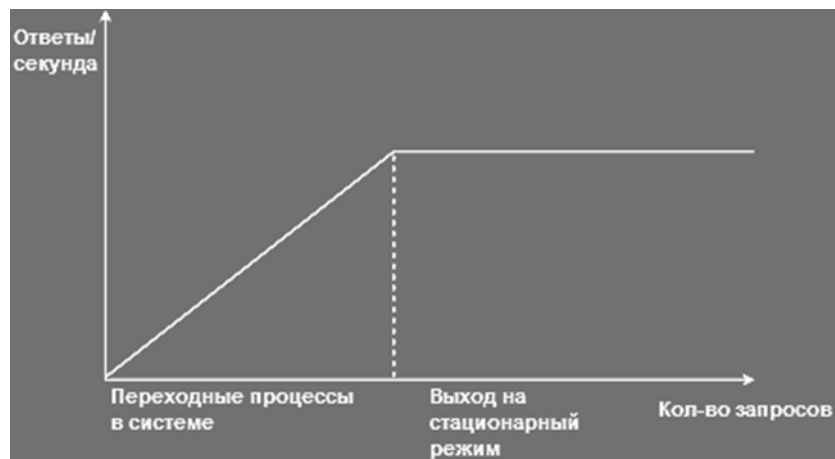


Источник: составлено авторами на основе данных [17; 18]

Рис. 8. Качественный анализ параметра среднего времени ответа на запрос

Начальные колебания исследуемого параметра подтверждают нашу гипотезу, что недостаточно исследовать стационарное поведение системы. Колебания графика свидетельствуют о так называемом «прогреве» процессора, связанном с накоплением статистики по его условным переходам.

Похожую ситуацию можно наблюдать при исследовании параметра скорости ответа на длинном промежутке времени (рис. 9).

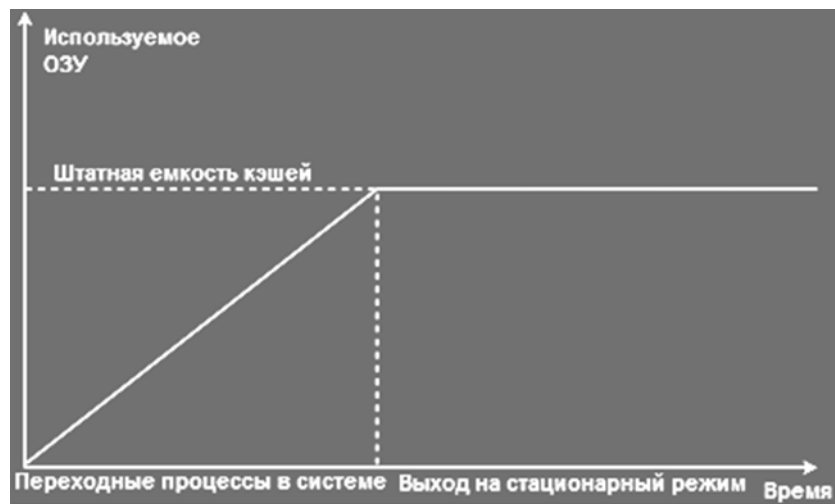


Источник: составлено авторами на основе данных [17; 18]

Рис. 9. Качественный анализ параметра скорости ответа на запрос

Из соображений энергоэффективности процессора данная характеристика начинает постепенно наращивать скорость работы. Данный эффект наиболее выражен у процессоров, выпускаемых компанией AMD. Данный эффект наиболее выражен у процессоров, выпускаемых компанией AMD.

Аналогичная зависимость наблюдается и при анализе такого параметра, как объем используемой оперативной памяти (рис. 10).

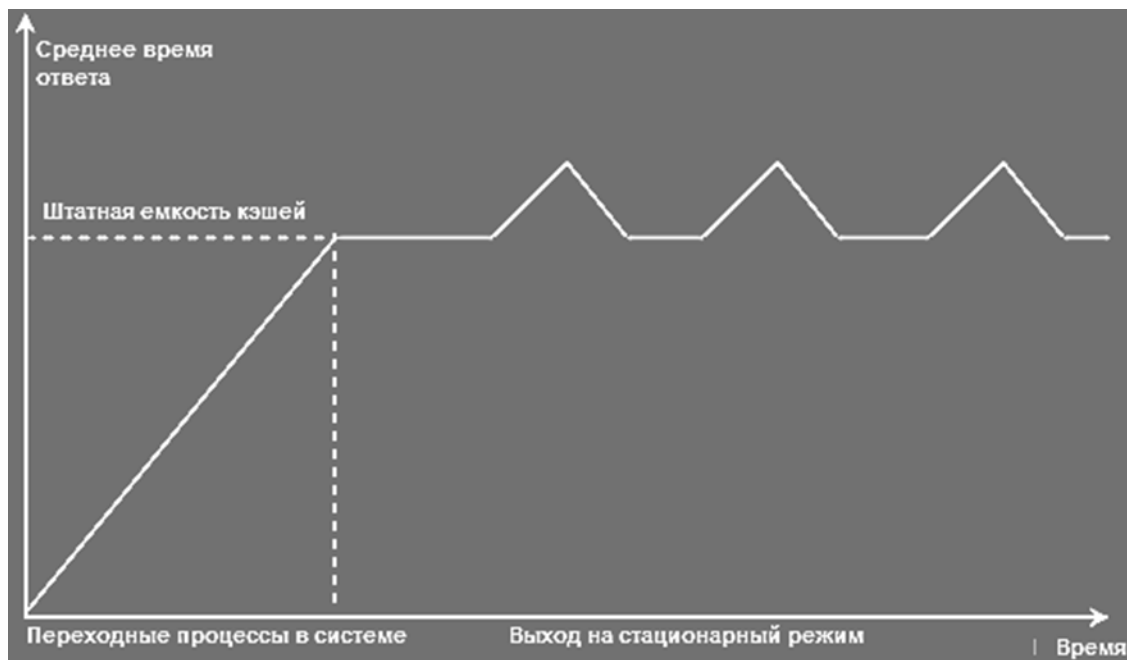


Источник: составлено авторами на основе данных [17; 18]

Рис. 10. Качественный анализ параметра объема оперативной памяти

Наконец, в завершение качественного анализа следует провести исследование характеристик изменения времени ответа в зависимости от общего времени работы РИС (рис. 11). Пики в области стационарного режима

свидетельствуют о наличии в системах такого рода периодических процессов, запускаемых по расписанию (всевозможные scheduling tasks, garbage collection и т.д.)



Источник: составлено авторами на основе данных [17; 18]

Рис. 11. Качественный анализ изменения времени ответа в зависимости от общего времени работы РИС

Заключение

Рассмотренные выше эффекты позволяют сделать ряд выводов в контексте разработки высоконагруженных распределенных систем. Главное состоит в том, что поверхностное тестирование общепринятыми инструментами не позволяет понять все проблемы и характеристики создаваемой системы. Наиболее эффективно проводить нагрузочное тестирование с помощью длительных по времени источников нагрузки (имитация запросов от пользователей). Количественные данные позволяют оценить длительность таких промежутков времени в 20-40 минут. Архитектурно обусловлено, что за это время практически все системы подобного класса выходят на штатную нагрузку и стационарный режим работы. Важным является тот факт, что характер поведения системы сильно

зависит от модели нагрузки на систему. Наиболее хрупкой и требующей внимательного рассмотрения оказалась открытая модель нагрузки. Это связано с тем, что в данном случае полностью отсутствует обратная связь между источником нагрузки и информационной системой. В условиях проводимой политики импортозамещения в области информационных технологий это является особенно важным, т.к. позволяет не только оценить принципиальную пригодность отечественного софта для решения тех или иных задач, но и, главное, оценить возможность плавности перехода на него без перерывов в организации бизнес-процессов компаний, что является главной оценкой качества исследуемых продуктов.

Список литературы

1. Ванина М.Ф., Давыдова Е.В., Ерохин А.Г., Фролова Е.А. Проблемы и перспективы использования российского и зарубежного свободного программного обеспечения в учебном процессе вуза // Методические вопросы преподавания инфокоммуникаций в высшей школе. – 2018. – Т. 7. – № 1. – С. 7-11.
2. Слядников П.Е., Тутова Н.В., Ерохин А.Г. Анализ проблем и перспектив интеграции блокчейна в системы Интернета вещей // Электронный научный журнал «Век качества». – 2025. – № 2. – С. 313-328. – URL: <https://www.agequal.ru/pdf/2025/225015.pdf>.
3. Дрожкин А.А. Анализ современных подходов к оценке качества программных продуктов // Вестник РЭУ им. Г.В. Плеханова. – 2022. – Т. 19. – № 6(126). – С. 207-218.
4. Бевзенко С.А. Основные виды и классификация нагрузочного тестирования // Вестник науки. – 2024. – Т. 4. – № 1(70). – С. 427-432.
5. Ярочкин В.И., Шевцова Т.А. Словарь терминов и определений по безопасности и защите информации. – М.: Ось-98, 1996. – 48 с.

6. Долгоруков А.Ю., Порай Д.С. Методика и инструментарий проведения нагрузочного тестирования // Труды ИСА РАН. – 2006. – Т. 23. – С. 174-189.
7. Московское метро переходит на Astra Linux и Postgres Pro // Cnews [Электронный ресурс]. – URL: https://www.cnews.ru/news/top/2025-10-23_moskovskij_metropolitan (дата обращения 29.10.2025).
8. McLaughlin B. Building Java Enterprise Applications. Vol. 1: Architecture. – O'Reilly, 2002. – 318 p. – URL: <https://theswissbay.ch/pdf/Gentoomen%20Library/Programming/Java/OReilly.Building%20Java%20Enterprise%20Applications%20Volume%20I.pdf> (дата обращения 29.10.2025).
9. Ванина М.Ф., Ерохин А.Г., Фролова Е.А. Применение облачных технологий в компаниях малого и среднего бизнеса // Век качества. – 2015. – № 1. – С. 61-64.
10. Ванина М.Ф., Ерохин А.Г., Фролова Е.А. Оценка внедрения облачных решений в бизнес компании // Век качества. – 2015. – № 2. – С. 29-33.
11. Java EE Version History and Important Changes [Электронный ресурс]. – URL: <https://www.codejava.net/java-ee/java-ee-version-history> (дата обращения 29.10.2025).
12. Сухомлин В.А. Методологический базис открытых систем // Открытые системы. – 1996. – № 4.
13. Бизнес. – М.:ЮНИТИ, 2000. – 356 с.
14. Джекобсон И. По кирпичику // Computerworld. – 1998. – № 42.
15. Рихтер Д. Программирование на платформе MS.NET Framework. – М.: Русская редакция, 2002. – 486 с.
16. Эккерсон В. В поисках лучшей архитектуры клиент-сервер // Сети. – 1995. – № 4.
17. Антонов А.В., Жарко Е.Ф., Промыслов В.Г. Проблемы оценки надежности и качества программного обеспечения в автоматизированных системах

управления технологическими процессами // Надежность. – 2015. – № 4. – С. 87-91.

18. Ермыкин А.Н. Разработка метода построения комплекса нагрузочного тестирования распределенной информационной системы: дис. ... канд. техн. наук: 05.13.13. – С.-Пб. гос. университет информационных технологий, механики и оптики. – С.-Пб, 2005. – 147 с.
19. Рогожников М. Анализ результатов нагрузочного тестирования [Электронный ресурс]. – URL: <https://habr.com/ru/company/tinkoff/blog/514314/> (дата обращения 17.11.2025).

Load Testing as a Method for Assessing of Software Quality in Distributed Information Systems under Import Substitution Conditions

Vanina Margarita Fedorovna

*Associate Professor, Candidate of Technical Sciences,
Associate Professor of the Department "Business Informatic",
Moscow Technical University of Communications and Informatics,
8A Aviamotornaya str., Moscow, 111024, Russia
m.f.vanina@mtuci.ru*

Erokhin Andrey Gustavovich

*Associate Professor, Candidate of Technical Sciences,
Associate Professor of the Department "Business Informatic",
Moscow Technical University of Communications and Informatics,
8A Aviamotornaya str., Moscow, 111024, Russia
a.g.erokhin@mtuci.ru*

Frolova Elena Aleksandrovna

*Senior Lecteur of the Department "Business Informatic",
Moscow Technical University of Communications and Informatics,
8A Aviamotornaya str., Moscow, 111024, Russia
e.a.frolova@mtuci.ru*

IT industry in Russia remains one of the most import-dependent sectors of the Russian economy. That is why the problem of import substitution is particularly acute here, and now an increasing number of companies from various industries are switching to domestic software. However, it is important not only to develop an information system based on domestic solutions, but also to ensure data migration to such a system without stopping business processes so that the new system can cope with all the loads. In other words, it is necessary to ensure the appropriate quality of domestic solutions so that they are not inferior or even superior to foreign ones. This is what the process of testing information systems under various loads designed to provide. In addition, in the modern world, it is becoming increasingly important to change the approach to building information systems: from traditional, static and centralized models to decentralized architectures. The basis of these new systems are data collection and analysis methods, which leads to the formation of distributed information systems (DIS). The article analyzes modern approaches to load testing processes in the context of the transition to domestic software and the construction of distributed information systems based on it.

Keywords: import substitution, software, distributed information systems, load testing, quality metrics, sanctions risks, digital sovereignty.