

УДК 517.929.2

DOI: [10.26102/2310-6018/2022.36.1.025](https://doi.org/10.26102/2310-6018/2022.36.1.025)

Моделирование обработки больших массивов данных в распределенных информационно-телекоммуникационных системах

Т.В. Мельникова¹✉, М.В. Питолин², Ю.П. Преображенский¹

¹ Воронежский институт высоких технологий, Воронеж, Российская Федерация

² Воронежский институт МВД России, Воронеж, Российская Федерация
tmelnikova910@gmail.com✉

Резюме: В статье рассматривается задача моделирования обработки больших массивов данных в распределенных информационно-телекоммуникационных системах. Проведен анализ проблемы, связанной с некорректной работой информационной системы. Показано, что распределенные системы позволяют распределить логику и данные по нескольким физическим серверам, что, при правильном распределении ресурсов и логики, дает возможность снизить риски сбоев. Показано, каким образом проводится расчет процента потери производительности от сбоя в одном из компонентов распределенной информационной системы. Рассмотрен фрагмент диаграммы вариантов использования информационной системы магазина. Если при построении информационной системы требуется закупить физические серверы, то при обеспечении условия одновременной работы серверов можно вычислить коэффициент уменьшения производительности информационной системы в случае сбоя на одном из серверов. Приведен график зависимости изменения производительности от количества узлов, составляющих распределенную сеть. Анализ показывает, что наибольший эффект по уменьшению рисков от добавления узлов в распределенную систему имеют случаи, когда серверов не слишком много. В случае, если распределенная система уже состоит из множества различных серверов и любая логика продублирована на многих из них, уменьшение рисков от добавления дополнительного сервера будет незначительным.

Ключевые слова: распределенная обработка данных, информационная система, моделирование, передача информации, алгоритм

Для цитирования: Мельникова Т.В., Питолин М.В., Преображенский Ю.П. Моделирование обработки больших массивов данных в распределённых информационно-телекоммуникационных системах. *Моделирование, оптимизация и информационные технологии*. 2022;10(1). Доступно по: <https://moitvvt.ru/ru/journal/pdf?id=1117> DOI: 10.26102/2310-6018/2022.36.1.025

Modeling of large data array processing in distributed information and telecommunication systems

T.V. Melnikova¹✉, M.V. Pitolin², Yu.P. Preobrazhenskiy¹

¹ Voronezh Institute of High Technologies, Voronezh, Russian Federation

² Voronezh Institute of the Ministry of Internal Affairs of Russia, Voronezh, Russian Federation
tmelnikova910@gmail.com✉

Abstract: The article considers the problem of modeling large data array processing in distributed information and telecommunication systems. The analysis of the problem, associated with the incorrect operation of the information system, is conducted. It is shown that distributed systems allow distribution

of logic and data across multiple physical servers, which, with the correct allocation of resources and logic, makes it possible to reduce the risks of failures. It is demonstrated how the calculation of productivity loss percentage from a failure in one of the distributed information system components is carried out. A fragment of the diagram, outlining the options for the store information system application, is examined. If it is necessary to purchase physical servers when building an information system, then it is possible to calculate the coefficient of decrease in the performance of the information system in the event of a failure on one of the servers while ensuring the conditions for simultaneous operation of servers. The graph, illustrating the dependence of the performance change on the number of nodes that comprise the distributed network, is given. The analysis indicates that the cases when there are not too many servers have the greatest effect on reducing the risks from adding nodes to a distributed system. If a distributed system already consists of many different servers and any logic is duplicated on many of them, the risk reduction from adding an additional server will be negligible.

Keywords: distributed data processing, information system, modeling, information transfer, algorithm

For citation: Melnikova T.V., Pitolin M.V., Preobrazhensky Yu.P. Modeling of large data array processing in distributed information and telecommunication systems. *Modeling, Optimization and Information Technology*. 2022;10(1). Available from: <https://moitvvt.ru/ru/journal/pdf?id=1117>
DOI: 10.26102/2310-6018/2022.36.1.025 (In Russ.).

Введение

Многие компании с развитием сети Интернет стали предоставлять свои услуги и сервисы людям по всей планете, что раскрыло свои недостатки сервисов, которые имеют один единственный сервер. Задачи таких компаний решают распределенные вычислительные системы, которые состоят из множества различных компьютеров, серверов и прочих устройств, каждое из которых выполняет свою роль в предоставлении сервиса клиенту.

Однако, распределенные системы, выполняя сложные задачи, имеют определенные недостатки и особенности, которые можно узнать в процессе эксплуатации или системного анализа [1].

Объектом исследования данной работы являются распределенные системы обработки данных. Предмет исследования – алгоритмы обработки данных в таких системах и особенности работы распределенных систем обработки данных.

Цель данной работы – рассмотреть проблемы и особенности обработки данных, проанализировать их работу в распределенных вычислительных системах, и предложить варианты их решения.

Проблемы обработки данных в распределенных системах

Распределенные вычисления – способ решения трудоемких вычислительных задач с использованием нескольких компьютеров, объединенных в параллельную вычислительную систему.

Распределенные вычислительные системы начинают быть необходимыми в момент, когда количество данных, которые необходимо обработать, превышает количество данных, которые можно обработать с помощью одного физического сервера. Распределенные информационные системы представляют собой совокупность программных и аппаратных решений.

Можно выделить несколько прикладных областей, где распределенные вычисления играют важную роль:

- Социальные сети
- Онлайн-игры
- Облачные сервисы обработки и хранения данных

– Поисквые сервисы

Архитектуры и принципы функционирования социальных сетей и систем хранения данных демонстрируют применение технологий и методов построения высоконагруженных распределенных сервисов.

В распределенных информационных системах очень многое зависит от среды передачи данных, достаточности ресурсов на выполнение необходимых операций над данными. Каналы связи в данный момент развиваются не очень быстро, поскольку упор идет уже даже не в несовершенство алгоритмов передачи данных, а скорее в физические ограничения среды передачи. Исследования ведутся, но на данный момент не существует способа быстрой передачи данных между устройствами, способного соревноваться с каналами передачи данных внутри компьютера [2].

Существует также и проблема грамотного распределения ресурсов между физическими устройствами распределенной вычислительной системы, поскольку каналы связи между отдельными компьютерами в вычислительной сети не настолько быстры, как каналы связи внутри одной вычислительной машины. В случае, если ресурсы будут размещены не оптимально, велика вероятность, что распределенная система не будет работать на полную мощность из-за возникновения узких мест в среде передачи данных.

Для решения этой проблемы следует дополнительно оптимизировать программное и аппаратное обеспечение под многопроцессорные и многомодульные аппаратные комплексы. Это требует больших трудозатрат, и в некоторых случаях целесообразнее будет разрабатывать изначально распределенную систему с поддержкой горизонтального масштабирования с минимально ощутимыми ограничениями.

Нехватка инструментов и средств разработки распределенных систем также является большой проблемой. Большинство разработок распределенных систем начинается с нуля, без применения каких-либо готовых компонентов, реже применяются готовые программные решения от сторонних разработчиков, и интегрируются между собой, но это все еще не очень автоматизированный процесс, требующий большого количества вложений и труда разработчиков. Решением данной проблемы могло бы стать создание универсального программно-инструментального средства для построения распределенных систем, что в целом упростит разработку, но не уберет полностью необходимость дополнительных вложений в разработку распределенной системы по сравнению с обычным приложением.

Кроме всего прочего, при разработке распределенного программного обеспечения необходимо моделирование распределенной системы, поскольку необходимо обеспечить защиту от воздействия внешних факторов, обеспечить конфиденциальность передаваемых между компонентами системы данных, а также в процессе эксплуатации необходимо будет поддерживать и отказоустойчивость системы. Моделирование работы распределенной системы можно произвести и на одном физическом сервере с помощью использования технологий виртуализации. Таким образом можно отладить взаимодействие компонентов и провести первичное нагрузочное тестирование для определения того, как система будет вести себя в ситуациях нехватки ресурсов, а также в прочих ситуациях, когда работа системы идет не совсем по плану.

Моделирование и тестирование (в том числе и нагрузочное) необходимы и при поддержке и доработке распределенной системы, чтобы вовремя выявлять появившиеся части кода, наиболее сильно использующие вычислительные ресурсы системы, а также возможные проблемы с нехваткой пропускной способности среды передачи данных [5].

Все описанные в данном разделе проблемы желательно решить в ближайшие годы, поскольку количество распределенных вычислительных систем растет с каждым днем, как и сложность алгоритмов обработки данных. Потому вопрос упрощения и увеличения прозрачности разработки распределенных информационных систем стоит очень остро уже в наше время.

Проблемы, связанные с некорректной работой информационной системы

Сбои могут возникать как со стороны программного, так и со стороны аппаратного обеспечения.

Аппаратное обеспечение может давать сбои при выходе из строя какого-либо из компонентов оборудования, входящего в состав аппаратного комплекса, на котором выполняется логика информационной системы, а также при непредвиденном отключении питания этого оборудования. Кроме того, с определенной периодичностью отдельные серверы требуется останавливать для нужд сервисного обслуживания (обновление операционной системы, установка обновления безопасности, и прочие действия, которые требуют либо перезагрузки операционной системы, либо перезапуска самой информационной системы).

Программные сбои возможны как по причине ошибок в программном коде, так и по причине невозможности выделения оперативной памяти под какие-либо нужды информационной системы по причине отсутствия ресурсов аппаратного обеспечения. Причем, программные ошибки могут быть как по причинам, связанным с бизнес-логикой (например, когда архитекторы и разработчики не учли каких-либо условий при проектировании и разработке информационной системы), так и по причинам, не связанным с ней (например, ошибки драйверов).

При этом, в случае сбоя требуется чаще всего перезапустить компонент, который дал сбой, либо сервер, если сбой аппаратный, или на уровне операционной системы. Кроме того, в таком случае требуется время на анализ причин происхождения сбоя, устранение этих причин и занесение в базу знаний способов решения подобных инцидентов [4].

Распределенные системы позволяют распределить логику и данные по нескольким физическим серверам, что, при правильном распределении ресурсов и логики, дает возможность снизить риски сбоев.

При этом сами по себе сбои в работе аппаратного и программного обеспечения могут быть частично спрогнозированы. Для этого существует мониторинг работы информационных систем, и различные системы мониторинга, обладающие самым разным функционалом от мониторинга конкретного параметра работы системы до предугадывания возможной ошибки ПО или аппаратного обеспечения заранее с соответствующим уведомлением администратору системы или специалисту технической поддержки.

Снижение рисков при отказах компонентов

Определение распределенной информационной системы предполагает, что система распределена по нескольким физическим либо виртуальным серверам.

Обычно распределенными называют информационные системы, логика которых распределена между несколькими различными серверами. В некоторых не особо крупных распределенных системах, где взаимодействующих компонентов больше, чем имеющих серверов, на каждом сервере может быть установлено по несколько приложений.

Из-за этого при аппаратном сбое может возникнуть такая ситуация, когда несколько приложений, участвующих в процессе обработки данных внутри единого потока, станут одновременно недоступны. Если какое-либо из приложений при этом будет находиться только на сервере, который дал сбой, то обработка данных в потоке будет приостановлена по причине отсутствия одного из компонентов на пути потока данных. При этом, в зависимости от того, каким образом происходит взаимодействие между приложениями, может случиться либо ошибка обработки данных одним из приложений распределенной системы, либо переполнение очереди сообщений в случае, если для временного промежуточного хранения данных используются очереди сообщений [5].

Исходя из этого, требуется постоянно мониторить такие параметры как количество запросов, обрабатываемых каждым обработчиком за единицу времени, либо длину очереди сообщений, в зависимости от способа взаимодействия приложений.

Распределенные системы по своей сути являются совокупностью электронных вычислительных устройств, работающих вместе для обеспечения работы каких-либо сервисов. В последние годы рынок процессоров в основном состоит из многоядерных моделей, причем производители стремятся в каждом новом поколении нарастить как число вычислительных ядер, так и производительность на каждое конкретное вычислительное ядро.

В процессорах применяют технологии многопоточности, позволяющие выполнять несколько потоков на одном вычислительном ядре, что тоже в целом помогает увеличить производительность системы [3, 6].

С увеличением производительности процессоров растут производительности других компонентов – оперативной памяти, накопителей данных. Однако, в любой вычислительной системе всегда можно найти узкое место, которое представляет собой ресурс, исчерпание которого происходит раньше остальных. В таких случаях можно как оптимизировать программное обеспечение для уменьшения потребления ресурса, так и установить, если это возможно, более производительный ресурс. Чаще всего применяется масштабирование ресурсов информационных систем, поскольку возможности оптимизации процессов не безграничны, и зачастую алгоритмы пишутся таким образом, чтобы получить баланс производительности и поддерживаемости программного кода. Это происходит потому, что аппаратное обеспечение стоит сильно дешевле времени, которое может уйти на лишнюю оптимизацию программного кода, и последующую его поддержку. Кроме того, после оптимизации скорости исполнения программный код может стать менее читаемым, что может привести к дополнительным трудозатратам на изучение алгоритма работы перед внесением необходимых правок.

Для анализа и уменьшения возможных рисков необходимо составить грамотную документацию по потоку обработки данных, в которой будет хорошо описан весь поток обработки данных, и точно указаны все компоненты распределенной системы, которые в обработке данных участвуют. Такая документация может сильно помочь при анализе возможных рисков, возникающих при отказе одного из программных или аппаратных компонентов [7].

Самый принципиально простой вариант распределенной информационной системы предполагает, что имеется количество серверов, равное количеству сервисов, которое нужно в итоге установить и настроить. В таком случае если для обработки какого-либо потока данных требуется задействовать все сервисы распределенной системы, то сбой любого одного из них означает остановку всей системы.

Таким образом можно понять, что наличие в распределенной системе нескольких обработчиков логики каждого из программных продуктов на разных серверах может быть полезным в случае сбоя одного из серверов, поскольку тогда производительность

системы уменьшится на процент, равный проценту того, какой процент вычислительной мощности предоставляет этот компонент, или все компоненты, на которые пришелся сбой. В то же время, если в информационной системе существует только один обработчик какой-либо логики, в случае сбоя в нем, весь поток обработки данных может быть остановлен. Процент потери производительности от сбоя в одном из компонентов распределенной информационной системы можно вычислить по следующей формуле:

$$\Delta P = 1 - \frac{P_{\text{общ}} - \sum P_{\text{комп}}}{P_{\text{общ}}}$$

Из этой формулы следует, что чем большее количество ресурсов попадает в так называемый домен сбоя, тем больше падает производительность информационной системы.

В распределенной информационной системе с дублированием ресурсов каждая копия одного из приложений может взаимодействовать с любой из копий другого приложения. Это позволяет в случае недоступности одного из серверов приложений быстро перераспределить нагрузку между оставшимися. Стоит заметить, что базы данных в этом случае должны постоянно синхронизироваться, поскольку в ином случае возможны ситуации, когда клиент получит некорректную информацию [8].

Для того, чтобы сбой части системы был минимально замечен для пользователей, требуется протестировать, какую нагрузку может создать предполагаемое количество пользователей на информационную систему в среднем и в часы пиковых нагрузок.

Этот вопрос можно исследовать уже после введения информационной системы в промышленную эксплуатацию, и более того, можно порекомендовать эксплуатирующим систему компаниям мониторить этот показатель, чтобы понимать, в какой момент количество необходимых для обработки прогнозируемого потока данных ресурсов приближается к количеству имеющихся ресурсов системы [9]. С помощью этого мониторинга можно также понять, в какой момент времени нагрузка на систему снижается. Это может позволить масштабировать ресурсы информационной системы в соответствии с текущими потребностями, что может позволить сэкономить затраты финансов, если пиковые моменты случаются слишком редко, а остальное время система обрабатывает в несколько раз меньше запросов, чем в пике.

Говоря о масштабировании ресурсов, нужно понимать, что разделяют горизонтальное и вертикальное масштабирование информационных систем. Вертикальное масштабирование предполагает увеличение производительности конкретного вычислительного устройства, или компонента системы, в то время как горизонтальное масштабирование заключается в масштабировании количества вычислительных блоков или компонентов. Стоит заметить, что вертикальное масштабирование информационных систем теряет актуальность, когда требуется количество вычислительных ресурсов, превышающее данный показатель у одного физического сервера. Кроме того, зачастую стоимость аппаратного обеспечения изменяется с ростом производительности нелинейно, то есть цена может расти больше, чем реальный прирост в скорости обработки данных. Это связано с тем, что разработка и производство более производительных компонентов [10] требует больших ресурсов, что отражается в ценовой политике. По этой причине в современных реалиях горизонтальное масштабирование выглядит наиболее подходящим решением для высоконагруженных систем.

Зная из мониторинга производительность одного экземпляра каждого вычислительного компонента информационной системы, можно вычислить количество таких компонентов, которое может пригодиться в определенный момент времени. Для

этого потребуются необходимая производительность системы и производительность одного компонента системы [11]. Для обеспечения резервирования, можно добавить в формулу еще и необходимый процент зарезервированных ресурсов на случай сбоя. Формула в общем виде выглядит следующим образом:

$$n_{ед} = \frac{P_{необх.}}{P_{един.}} \times (1 + a).$$

В данной формуле a означает процент ресурсов, который должен быть зарезервирован в информационной системе в нужный момент времени на случай увеличения нагрузки, либо возможного сбоя какой-либо из копий нужного приложения. Исходя из этой формулы, чем больший процент резервирования требуется, тем больше вычислительных ресурсов необходимо.

Стоит заметить, что перед проектировщиками информационных систем часто стоит и задача подбора оптимальной конфигурации аппаратного обеспечения для сервера. В данном случае мы встречаем пересечение сфер применения вышеописанных формул, и получим задачу оптимизации ресурсов, которая дополнится еще одним фактором – целевой ценой информационной системы.

Анализ вариантов использования информационной системы

Рассмотрим часть диаграммы вариантов использования информационной системы (Рисунок 1).

На этой диаграмме можно увидеть, какие задачи каждого из типов пользователей должна решать информационная система. Исходя из этого система делится на компоненты, которые располагаются таким образом, чтобы целевым пользователям было максимально удобно пользоваться ресурсами этой информационной системы, и чтобы для пользователей информационная система работала наиболее стабильно.

Для начала стоит понять, как должна будет выглядеть информационная система сети магазинов. Правильнее всего в каждом магазине установить собственный сервер для обработки собственных данных, а также спроектировать систему сбора общей информации от всех магазинов с функционалом аналитики и формирования отчетов.

Для современных распределенных информационных систем можно порекомендовать микросервисный подход к разработке информационной системы. Данный подход позволяет, в случае необходимости, изменять реализацию каких-либо компонентов информационной системы компании, не меняя логики остальных элементов. Кроме того, информационные системы с микросервисной архитектурой довольно неплохо поддаются масштабированию и распределению приложений по большому количеству серверов. Это очень полезно для информационных систем. Однако данный подход применим далеко не для каждой задачи.

Для предупреждения потери данных в информационных системах, занимающихся потоковой обработкой данных применяют очереди сообщений. С точки зрения внутренней логики – очередь сообщений – хранилище данных, в котором хранятся различные сообщения, которые можно в данную очередь записывать, хранить там до востребования и считывать. Очереди сообщений существуют для решения проблемы нехватки ресурсов информационной системы в периоды пиковых нагрузок.

Однако, к построению архитектуры и логики центрального сервера сети магазинов может быть несколько подходов. Самый очевидный и простой – разместить единственное общее приложение на центральном сервере. Данный подход имеет право

на жизнь при сравнительно небольших нагрузках и ровно до того момента, пока группе компаний не начинает требоваться непрерывная работа сервера.

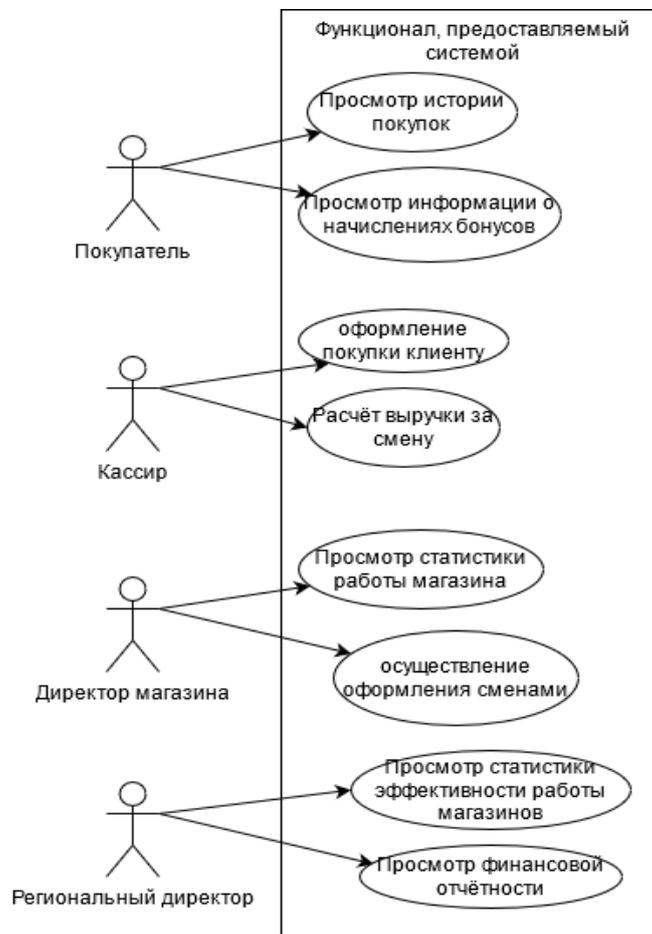


Рисунок 1 – Фрагмент диаграммы вариантов использования информационной системы магазина

Figure 1 – Fragment of a diagram showing the options for the store information system application

С ростом количества клиентов сеть пользователей информационной системы будет расширяться, и для этого потребуются решения, необходимую производительность работы которых будет сложно поместить на одном физическом сервере. Кроме того, зачастую требуется высокая доступность информационной системы.

Таким образом, мы приходим к тому, что в крупных компаниях для собственных приложений следует использовать микросервисную архитектуру с распределением приложений по различным виртуальным и физическим машинам. Микросервисная архитектура позволяет «раздробить» крупную систему на составные части, каждая из которых будет выполнять одну конкретную задачу, заранее ей предзаданную.

При этом, если при построении информационной системы требуется закупить физические серверы, то при обеспечении условия одновременной работы серверов можно вычислить коэффициент уменьшения производительности информационной системы в случае сбоя на одном из серверов.

Данное значение можно вычислить по следующей формуле:

$$\Delta v = \frac{1}{x} \times 100$$

В данной формуле x – количество серверов, составляющих информационную систему, Δv – изменение производительности системы. Данная формула показывает, что чем больше в системе серверов, несущих в себе всю возможную логику обработки данных, тем информационная система несет меньше рисков при деградации либо сбое на одном из серверов.

График зависимости изменения производительности от количества узлов, составляющих распределенную сеть, расположен на Рисунке 2:

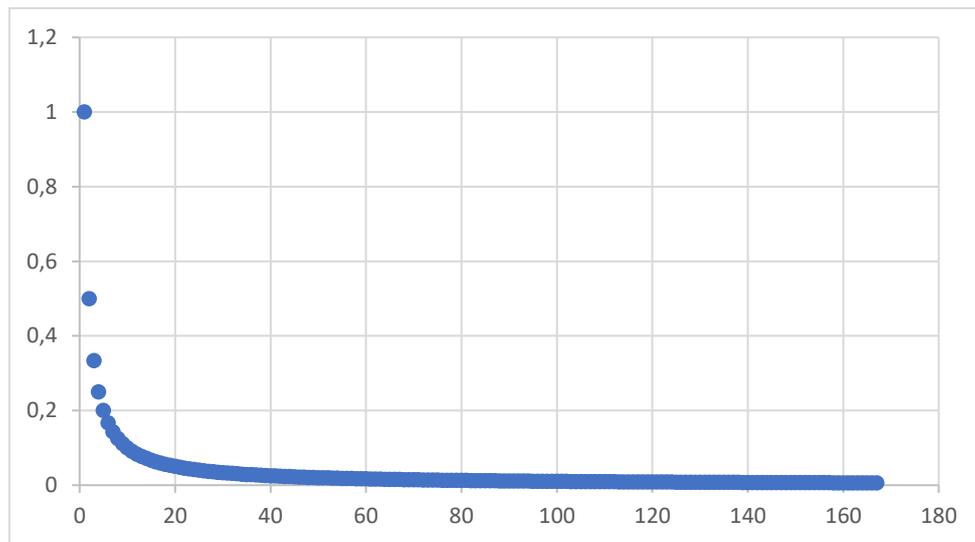


Рисунок 2 – Зависимость критичности сбоя одного узла для распределенной системы от количества узлов

Figure 2 – Dependence of a single node failure criticality for a distributed system on the number of nodes

Исходя из этой диаграммы, наибольший эффект по уменьшению рисков от добавления узлов в распределенную систему имеют случаи, когда серверов не слишком много. В случае, если распределенная система уже состоит из множества различных серверов и любая логика продублирована на многих из них, уменьшение рисков от добавления дополнительного сервера будет незначительным.

Однако, добавление дополнительных серверов в распределенную систему увеличит производительность этой системы.

Одним из главных недостатков при использовании распределенных вычислительных систем называют сложность администрирования таких систем. Однако, сложность администрирования может уменьшиться при автоматизации администрирования таких систем. Для этого во многих компаниях существуют направления DevOps, занимающиеся автоматизацией доставки обновлений и администрирования информационных систем.

Кроме того, существует множество облачных сервисов, которые предоставляют инструменты для простого управления приложениями в облаке. Это в том числе может позволить сократить расходы за счет избавления от необходимости поддерживать собственные серверы, а также оптимизировать оставшуюся часть расходов за счет

масштабирования количества запущенных экземпляров обработчиков логики в зависимости от текущих потребностей.

Заключение

Распределенные информационные системы позволяют решать задачи, которые было бы невозможно решить за короткий промежуток времени на отдельно стоящем сервере. Для обеспечения доступности сервиса даже при наличии неполадок на каком-либо из узлов применяют репликацию – дублирование логики и данных между различными серверами. Однако, с увеличением количества серверов, занимающихся обработкой определенной логики, возникновение ошибки на одном из серверов не особо влияет на производительность всей информационной системы в целом. В ходе работы были рассмотрены меры по оптимизации и вычислению оптимального количества узлов обработки данных в информационной системе.

СПИСОК ИСТОЧНИКОВ

1. Клименко Ю.А., Преображенский А.П. О возможностях использования теории перколяции при анализе распределительных электрических сетей. *Вестник Воронежского института высоких технологий*. 2021;2(37):57–59.
2. Преображенский Ю.П., Аветисян Т.В., Ружицкий Е. Особенности функционирования систем интернета вещей. *Вестник Воронежского института высоких технологий*. 2021;4(39):71–74.
3. Львович Я.Е., Мельникова Т.В., Преображенский А.П. Оптимизация беспроводных систем внутри помещений. *Вестник Воронежского института высоких технологий*. 2021;4(39):61–63.
4. Исичко В.Д., Линкина А.В. Аспекты обеспечения кибербезопасности в условиях цифровой трансформации. *Вестник Воронежского института высоких технологий*. 2021;4(39):48–51.
5. Львович Я.Е., Мельникова Т.В., Преображенский А.П. Анализ ключевых особенностей машинного обучения. *Вестник Воронежского института высоких технологий*. 2021;4(39):34–36.
6. Львович Я.Е., Карлин П.В., Преображенский Ю.П. Об особенностях моделирования беспроводных сенсорных сетей. *Вестник Воронежского института высоких технологий*. 2021;4(39):20–23.
7. Преображенский Ю.П., Чупринская Ю.Л., Кравцова Н.Е. Анализ характеристик, используемых при проектировании беспроводных систем связи. *Вестник Воронежского института высоких технологий*. 2020;3(34):47–49.
8. Филипова В.Н. О применении информационных технологий в туристической сфере. *Успехи современного естествознания*. 2012;6;10(4):830–834.
9. Lvovich I.Ya., Preobrazhensky A.P., Choporov O.N. Analysis of potential of error-correcting capabilities of codes. *Life Science Journal*. 2013;10(4):830–834.
10. Мельникова Т.В., Воробьева В.В., Ружицкий Е. Анализ некоторых экспериментальных подходов для беспроводных систем связи. *Вестник Воронежского института высоких технологий*. 2020;3(34):34–37.
11. Клименко Ю.А., Преображенский А.П. Особенности методов многомерного анализа. *Вестник Воронежского института высоких технологий*. 2020;4(35):44–46.
12. Львович Я.Е., Львович И.Я., Преображенский А.П., Головинов С.О. Исследование методов оптимизации при проектировании систем радиосвязи. *Теория и техника радиосвязи*. 2011;1:5–9.

REFERENCES

1. Klimenko YU.A., Preobrazhenskij A.P. O vozmozhnostyah ispol'zovaniya teorii perkolyacii pri analize raspredelitel'nyh elektricheskikh setej. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2021;2(37):57–59. (In Russ.)
2. Preobrazhenskij YU.P., Avetisyan T.V., Ruzhickij E. Osobennosti funkcionirovaniya sistem interneta veshchej. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2021;4(39):71–74. (In Russ.)
3. L'vovich YA.E., Mel'nikova T.V., Preobrazhenskij A.P. Optimizaciya besprovodnyh sistem vnutri pomeshchenij. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2021;4(39):61–63. (In Russ.)
4. Isichko V.D., Linkina A.V. Aspekty obespecheniya kiberbezopasnosti v usloviyah cifrovoj transformacii. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2021;4(39):48–51. (In Russ.)
5. L'vovich YA.E., Mel'nikova T.V., Preobrazhenskij A.P. Analiz klyuchevykh osobennostej mashinnogo obucheniya. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2021;4(39):34–36. (In Russ.)
6. L'vovich YA.E., Karlin P.V., Preobrazhenskij Yu.P. Ob osobennostyah modelirovaniya besprovodnyh sensoryh setej. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2021;4(39):20–23. (In Russ.)
7. Preobrazhenskij Yu.P., CHuprinskaya Yu.L., Kravcova N.E. Analiz harakteristik, ispol'zuemyh pri proektirovanii besprovodnyh sistem svyazi. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2020. 3 (34):47–49. (In Russ.)
8. Filipova V.N. O primenenii informacionnyh tekhnologij v turisticheskoy sfere. *Uspekhi sovremennogo estestvoznaniya = Advances in current natural sciences*. 2012;6:112–113. (In Russ.)
9. Lvovich I.Ya., Preobrazhensky A.P., Choporov O.N. Analysis of potential of error-correcting capabilities of codes. *Life Science Journal*. 2013;10(4):830–834. (In Russ.)
10. Mel'nikova T.V., Vorob'eva V.V., Ruzhickij E. Analiz nekotoryh eksperimental'nyh podhodov dlya besprovodnyh sistem svyazi. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2020;3(34):34–37. (In Russ.)
11. Klimenko YU.A., Preobrazhenskij A.P. Osobennosti metodov mnogomernogo analiza. *Vestnik Voronezhskogo instituta vysokih tekhnologij*. 2020;4(35):44–46. (In Russ.)
12. L'vovich YA.E., L'vovich I.YA., Preobrazhenskij A.P., Golovinov S.O. Issledovanie metodov optimizacii pri proektirovanii sistem radiosvyazi. *Teoriya i tekhnika radiosvyazi*. 2011;1:5–9. (In Russ.)

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Мельникова Томара Вениаминовна, студент Воронежского института высоких технологий, Воронеж, Российская Федерация.
e-mail: tmelnikova910@gmail.com

Melnikova Tomara Veniaminovna, student of the Voronezh Institute of High Technologies, Voronezh, Russian Federation.

Питолин Михаил Владимирович, кандидат технических наук, доцент, Воронежский институт МВД России, Воронеж, Российская Федерация.
e-mail: pmv_m@mail.ru

Pitolin Mikhail Vladimirovich, Candidate of Technical Sciences, Associate Professor, Voronezh Institute of the Ministry of Internal Affairs of Russia, Voronezh, Russian Federation.

Преображенский Юрий Петрович, профессор Воронежского института высоких технологий, Воронеж, Российская Федерация.
e-mail: petrovich@vvt.ru

Preobrazhensky Yuri Petrovich, Professor, Voronezh Institute of High Technologies, Voronezh, Russian Federation.

*Статья поступила в редакцию 28.12.2021; одобрена после рецензирования 21.03.2022;
принята к публикации 30.03.2022.*

*The article was submitted 28.12.2021; approved after reviewing 21.03.2022;
accepted for publication 30.03.2022.*