

С.К. Джолдасбаев¹, Б.О. Куламбаев²

¹ *Казахский национальный университет им. аль-Фараби, г. Алматы, Казахстан*

² *Университет «Нархоз», г. Алматы, Казахстан*

ПРИМЕНЕНИЕ АЛГОРИТМОВ БАЛАНСИРОВКИ НАГРУЗКИ ДЛЯ ПОВЫШЕНИЯ КАЧЕСТВА ПРЕДОСТАВЛЕНИЯ УСЛУГ

Аннотация

Для развития решений задач обеспечения высокоскоростного Интернета, то есть качественным сервисом обслуживания, до определенного момента существует возможность повышения качества за счет увеличения аппаратных ресурсов системы, но, как показывает практика, не всегда количество означает качество, и эффективность системы предоставления услуг учитывает именно выгодное позиционирование ресурсов с алгоритмическими балансировками нагрузки на серверах с максимальной пользой, как для пользователя, так и для стороны, предоставляющей услуги.

В статье предоставлены результаты исследования и анализа алгоритмов балансировки, методов реализации для распределения нагрузки на серверах и повышения качества предоставления услуг. Исследования в данном направлении являются весьма актуальными и востребованными, в статье приводятся анализ и описание методов статических и динамических решений, преимущества и недостатки алгоритмов.

Ключевые слова: повышения качества предоставления услуг, балансировка нагрузки, алгоритмы балансировки.

Аңдатпа

С.К. Джолдасбаев¹, Б.О. Куламбаев²

¹ *Әль-Фараби атындағы Қазақ ұлттық университеті, Алматы қ., Қазақстан*

² *«Нархоз» Университеті, Алматы қ., Қазақстан*

ҚЫЗМЕТ КӨРСЕТУ САПАСЫН АРТТЫРУ ҮШІН ЖҮКТЕУДІ ҮЛЕСТІРУ АЛГОРИТМДЕРІН ҚОЛДАНУ

Жоғарыжылдамдықты Интернетті қамтамасыз ету шешімдерін, яғни сапалы қызмет көрсету сервері, дамыту үшін белгілі уақытқа дейін жүйенің аппараттық ресурстарын арттыру арқылы жүргізуге болмақ, алайда, тәжірибе көрсеткендей, бұл әрқашанда оңтайлы әдіс болып табылмайды, және қызмет көрсету жүйелерінің оңтайлығы ресурстарды үлестіруді реттеу алгоритмдерімен сәйкес тиімді позициялау тұтынушы мен қызмет көрсетуші жақ үшін де тиімді болатынын көрсетеді.

Ұсынылған жұмыста серверлердегі жүктеуді реттеу алгоритмдерін, әдістерін зерттеу мен талдау және қызмет көрсету сапасын арттыру нәтижелері ұсынылған. Бұл бағытта зерттеулер оңтайлы және сұранысқа ие болып табылады. Жұмыста статикалық және динамикалық шешімдер әдістері, алгоритмдердің артықшылықтары мен кемшіліктері көрсетіледі.

Түйін сөздер: қызмет көрсету сапасын арттыру, жүктеуді үлестіру, үлестіру алгоритмдері.

Abstract

APPLICATION OF LOAD BALANCING ALGORITHMS FOR IMPROVING THE QUALITY OF SERVICE

Joldasbayev S.¹, Kulambayev B.²

¹ *Al-Farabi Kazakh National University, Almaty, Kazakhstan*

² *«Narхоз» University, Almaty, Kazakhstan*

To develop solutions to the problems of providing high-speed Internet, that is, a high-quality service, up to a certain point, there is the possibility of improving quality by increasing the hardware resources of the system, but, as practice shows, quantity does not always mean quality, and the effectiveness of the service delivery system takes into account the advantageous positioning of resources with algorithmic load balancing on servers with maximum benefit, both for the user and for the party providing services.

This article provides the results of research and analysis of balancing algorithms, implementation methods for load balancing on servers and improving the quality of service delivery. Research in this direction is very relevant and in demand, the article provides an analysis and description of static and dynamic solutions, the advantages and disadvantages of algorithms.

Keywords: QoS, load balancing, balancing algorithms.

Введение

В настоящее время использование Интернета стремительно растет по всему миру. Так, в Казахстане впервые настоящий Интернет появился еще 1997 году и стоил \$10 за час, с сервисом в среднем 14 400 бит/сек. Интернет был роскошью, ее доступность и скорость со временем повысилась, но тем не менее во многих регионах страны до сих пор оставляет желать лучшего, даже в радиусе крупных мегаполисов. Будет правильно, если мы отметим, что большую роль сыграл проект ОАО «Казахтелеком» – «Зона Интернет».

В конце 2000 года в Казахстане было зарегистрировано 1 945 сайтов, тогда как сейчас за одну неделю появляются 8-12 новых сайтов, а количество ресурсов и пользователей по всему Миру удваивается примерно за один год.

Постоянное увеличение объемов и ресурсов при использовании Интернета приводит к тому, что для многих сервисов важна возможность стабильной работы при больших нагрузках, так как для многих конкурентных фирм, пользующихся тем или иным сервисом, это играет значимую роль во всех производственных вопросах, вплоть до миграции клиентов к конкурентам.

И неудивительно, что многие предпочитают пользоваться услугами продвинутых в этой области крупных компаний, например, Amazon, Google и т.д., которые используют кластеры серверов как средство развертывания приложений и в качестве балансировщиков нагрузки.

Такие кластеры серверов (далее просто серверы) дают возможность не беспокоиться о сбоях системы – грамотно устроенные балансировщики нагрузки реализуют оптимальное управление поступающих запросов к серверам, что способствует реализации равномерной нагрузки на узлы, понижает потери производительности и обеспечивает максимально возможное время ответа на запрос.

На сегодняшний день разработано множество алгоритмов балансировки нагрузки на серверах, но не все алгоритмы применяются на практике. В основном, многие алгоритмы работают с учетом нагрузки на определенный сервер (кластеры серверов), принимая в расчет только его вычислительную мощность. Во многих случаях тестирование разработок проводится в гомогенных системах.

Обычно крупные сервисы развернуты на кластерах, состоящих из множества, в частности, гетерогенных узлов. Балансировка нагрузки в таких сервисах является актуальной задачей, так как не существует универсального решения для всех серверов.

Тем не менее, с целью расширения возможностей таких систем применимы определенные алгоритмы балансировки.

Обзор трудов

Большое количество известных ученых, например, Клейнрок, С. Блейк, Д. Гроссман, З. Ван, Стеклов В. К., Беркман Л. Н., а также такие исследовательские центры, как мобильные ad-hoc Сети, Internet Engineering Task Force, Center for Embedded Networked Sensing занимаются управлением и распределением трафика.

Однако, несмотря на огромное количество публикаций и усилия производителей, задача построения моделей движения, наилучшим образом отражающих его функционирование в реальных условиях, до сих пор не решена [1-5].

В частности, поскольку обучение с подкреплением предоставляет потенциал для разработки оптимальной политики распределения без явного знания модели путем изучения последствий каждого действия, существующие работы по алгоритмам ML в основном сосредоточены на обучении с подкреплением [5, 6]. Для их изучения не требуется ни явной модели системы, ни явной модели трафика. RL относится к процессу обучения, в котором обучающийся агент может научиться принимать соответствующие решения посредством взаимодействия с внешней средой [3].

В частности, помимо агента обучения и окружающей среды, система обучения с подкреплением состоит из политики, функции вознаграждения и функции ценности.

Пусть S -множество состояний окружающей среды, а A -множество действий, соответственно.

Другим популярным алгоритмом машинного обучения является машина опорных векторов (SVM). Он широко применяется в различных областях, таких как распознавание образов, классификация и интеллектуальный анализ данных.

Однако SVM не являются предпочтительными в онлайн-приложениях, поскольку сложность обучения и тестирования стандартных SVM составляет $O(nm + m^3)$ и (m) соответственно, где n -размер данных, а m -число опорных векторов. С другой стороны, для снижения сложности были предложены некоторые приближенные методы [9].

Например, [10] уменьшает сложность до $O(nd \max)$, где $d \max$ -число выбранных базисных функций.

Для решения задачи управления ресурсами было предложено несколько работ по алгоритмам машинного обучения [4-9]. Для контроля допуска [7] был получен сложный набор правил, который может быть использован для определения оптимальной конфигурации ненаблюдаемой рабочей нагрузки на основе алгоритмов машинного обучения.

[9] применил RL для автоматической настройки параметров в многоуровневых веб-системах, где восемь параметров на веб-уровне и уровне приложений выбираются так, чтобы они состояли из пространства состояний.

Для каждого параметра существует три возможных действия: увеличение, уменьшение и сохранение.

Политика основана на методе электронной жадности. Чтобы подавить низкую производительность из-за плохой инициализации, они предложили алгоритм для построения различных политик инициализации для различных сценариев.

Для масштабирования виртуальных машин [8] была предложена методика обучения итерационной модели на основе искусственной нейронной сети (ANN) для прогнозирования потребности в вычислительных ресурсах в виртуальных средах.

[7] применил RL для обучения нелинейных аппроксиматоров (например, многослойных перцептронов) вместо таблицы поиска для горизонтального масштабирования VM, где состояние определяется как скорость поступления запроса, а действие заключается в определении количества выделенных серверов.

Поскольку на практике пространство состояний экспоненциально растет с увеличением числа параметров, авторы применили аппроксиматор нелинейных функций в качестве внешней политики, чтобы избежать плохой производительности, которая ожидалась бы при онлайн-обучении. В последнее время в работах [5-8] было предложено несколько работ по нечеткому управлению для управления ресурсами.

В работе [8] контроль допуска осуществляется нечетким управлением с целью управления QoS, где параметр поворота Maxclients в каждом интервале управляется нечетким контроллером.

Для масштабирования VM [5] предпринята попытка отразить нелинейное поведение в использовании ресурсов VM путем разработки нечеткой модели оценки.

Подход делится на два этапа. Во-первых, метод моделирования на основе нечеткой логики используется для изучения поведения системы, не требуя каких-либо предварительных знаний. Затем прогностический контроллер прогнозирует потребность в ресурсах всех виртуальных машин и выполняет действия, основанные на этой модели.

[6] предложил нейронный нечеткий контроллер для основанной на процентилях сквозной гарантии задержки через виртуализированный многоуровневый серверный кластер, где гауссовские функции принадлежности сначала используются для размывания среднего времени обслуживания, s_i , и дисперсии времени обслуживания, σ_i , распределения запросов на уровне i , соответственно.

Затем нечеткая нейронная сеть применяется для онлайн-обучения на этапе вывода.

Кроме того, для дальнейшего повышения производительности вводится коэффициент масштабирования выходных данных. Он не зависит от модели и способен адаптировать параметры управления с помощью быстрого онлайн-обучения.

По сравнению с другими контролируруемыми методами машинного обучения, он не требует автономного обучения.

Анализ методов разработки алгоритмов балансировки

С точки зрения эффективности алгоритм считается хорошим, если удовлетворяет определенным требованиям, допустимым в пределах работы в режиме реального времени.

Например, если алгоритм позволяет системе предоставлять возможность горизонтального масштабирования, продолжать работу при выходе из строя некоторых узлов, то есть быть отказоустойчивым.

Методы разработки алгоритмов балансировки [8, 9] хоть и имеют разные подходы, соответствуют следующим требованиям:

1. Предсказуемость.
2. Равномерная или справедливая загрузка ресурсов системы.
3. Масштабируемость.

Во многих работах на сегодняшний день делают акцент как основные алгоритмы балансировки, имеющие наибольшее практическое применение, такие алгоритмы, как Round Robin, Weighted Round Robin, Least Queue, Load Least, Sticky session, алгоритмы группы Least Connections (Least Connections, Locality-Based Least Connection Scheduling, Locality-Based Least Connection Scheduling with Replication Scheduling) [5-10].

Используя следующие обозначения свойств алгоритмов, попытаемся дать их подробное описание:

ω_i – интенсивность обслуживания,

$p_i = \lambda_i / \lambda$ – вероятность направления запроса на i -й сервер,

$\lambda_i = \phi_i + \sum_{j=1}^n x_{ji}$ – интенсивность потока заявок поступающего на i -й сервер,

$\rho_i = \lambda_i / \omega_i$ – загруженность i -го сервера.

1) Round Robin (RR) – распределение заявок происходит по очереди, от первого до конечного циклический, все серверы получают в среднем одинаковое число заявок:

$$p_i = \frac{1}{n} = const, \lambda_i = \frac{\lambda}{n}, T = \sum_{i=1}^n \frac{1}{n\omega_i - \lambda}.$$

2) Weighted Round Robin(WRR) – распределение заявок по порядку, при условии, что каждому серверу присваивается весовой коэффициент в зависимости от производительности и мощности узла, и заявки на них поступают соответственно с принятыми правилами:

$$\frac{p_i}{w_i} = const, T = \sum_{i=1}^n \frac{1}{\frac{w}{w_i} \omega_i - \lambda}, \quad w = \sum_{i=1}^n w_i.$$

3) Least Queue – динамический алгоритм с обратной связью, заявка будет направлена к серверу, с наименьшим числом заявок на момент, в таком порядке длина очереди на всех серверах будет одинаковой:

$$\sum Q_i = \frac{\rho_i^2}{1 - \rho_i} = const, i = \min_{j=1, n} \{j: \lambda < \lambda^j\},$$

$$p_j = \frac{i\omega_j - \omega_\Sigma^i}{i\lambda} + \frac{1}{i}, \Leftarrow \lambda > \lambda^j = \omega_\Sigma^{j-1} - (j-1)\omega_j,$$

$$p_j = 0, \Leftarrow \lambda < \lambda^j,$$

$$T = \frac{i}{\omega_\Sigma^i - \lambda}$$

4) Least Load – динамический алгоритм с обратной связью. Заявка направляется на тот сервер, который менее всего загружен. Величина загруженности сервера может определяться, например, по времени соединения с сервером. Нагрузка на все серверы одинакова:

$$1 - U_i = \rho_i = const, p_i = \frac{\omega_i}{\omega_\Sigma}, T = \frac{n}{\omega_\Sigma - \lambda}$$

5) Least Connections – динамический алгоритм с обратной связью, с учетом количества подключений, поддерживаемых серверами в текущий момент времени. Заявка направляется на тот сервер, который менее всего загружен.

Таблица-1. Сравнительные характеристики алгоритмов балансировки

Название	Описание	Преимущества	Недостатки
Round Robin	перебор по круговому циклу	независимость от протокола, стоимость реализации, отсутствие связи меж серверами	однородность ресурсов, отсутствие информации о загруженности
Weighted Round Robin	перебор по круговому циклу с учетом весов серверов	гибкое распределение нагрузки, эффективность при известности состава серверов в кластере	предварительное определение производительности и мощности серверов
Least Queue	заявка направлена к серверу с наименьшим числом заявок на момент	длина очереди на всех серверах одинаковая	не учитывает загруженности отдельных запросов
Least Load	заявка направляется на менее загруженный сервер	нагрузка на все серверы одинакова	
Least connections	запросы передаются серверу с наименьшим количеством активных подключений.	надежность и повышение отказоустойчивости за счет подачи запроса менее загруженному узлу, стоимость, отсутствие необходимости данных о составе серверов	не учитывает загруженности отдельных запросов
Weighted Least Connections	учитывает при распределении нагрузки количество активных подключений и весовой коэффициент серверов	определение загруженности узла и учитывает весовой коэффициент серверов	не учитывает загруженности отдельных запросов
Least Connections, Locality-Based Least Connection Scheduling	принцип LC + за каждым из клиентских серверов закрепляется группа клиентских IP запросы кот. направляются на главный сервер, если он загружен перенаправляет запрос на другой сервер	Эффективен для для кэширующих прокси-серверов	не учитывает загруженности отдельных запросов, требует дополнительных ресурсов
Locality-Based Least Connection Scheduling with Replication Scheduling	каждый IP-адрес или группа IP-адресов закрепляется за группой серверов запрос передаётся наименее загруженному серверу из группы если все серверы из главной группы перегружены будет зарезервирован новый сервер	позволяет избежать избыточной репликации	требует дополнительных инструкции и энергетических затрат во время пиковой нагрузки
Sticky session	запросы поступают к серверу кластера, на который был направлен запрос при создании сессии	независимость от протокола, отсутствие связи меж серверами, поддержка в вебсервере NGINX	Не учитывается нагрузку на конкретный сервер при распределении

Рассмотренные алгоритмы можно эффективно применять для балансировки нагрузки на серверах при определенных условиях.

Модификация алгоритма Least Connections

На основании проведенных исследований обеспечение качества распределения нагрузки на серверах реализовано следующими алгоритмами: Round Robin, Weighted Round Robin, Load Least, Least Connections. Проведенные теоретические и вычислительные исследования показали преимущества и недостатки данных алгоритмов балансировки нагрузки на серверах развернутых в кластере «single-instance application». В настоящее время актуальным является алгоритм балансировки Least Connections, который в частности, используется для сервисов, развернутых в кластере «single-instance application» - на каждом из узлов имеется свой экземпляр приложения, в качестве хэш-таблицы используется распределенный кэш, данные в котором доступны на всех серверах [11].

Для улучшения исходного алгоритма предложена модификация, где используется не только количество активных подключений, но и определенный приоритет к серверу в зависимости от его ресурсов (мощности) по сравнению других в системе. Преимуществом данного алгоритма является и возможность инициализации новых узлов кластера, не только из файла с настройками кластера, но и по мере получения новых запросов. В случае если в настройках сервера не находится узел, к которому обращается запрос, то параметры запрашиваемого узла сохраняются и обеспечивается возможность динамического расширения состава серверов (кластеров), и данному узлу назначается самая минимальная нагрузка, так как вычислительные мощности данного сервера неизвестны.

Предположим, кластер состоит из N количества серверов. В зависимости от поступающих запросов кластер серверов предоставляет разные количества узлов: $S = \overline{1, N}$. В первую очередь алгоритм делает определение наличия параметров целевого сервера из запроса пользователей сравнивая ее с хэш-таблицей, где хранятся данные серверов в кластере S_i . Если идентичный сервер не находится, заносится запись нового узла S_{i+1} . Следом идет определение используемых серверов:

а) если используется только один сервер $S = 1$, то текущий запрос будет направлен к данному серверу,

б) если количество больше двух $S > 2$, то сортируется список серверов, зависящие от количества активных подключений и весов серверов, предусмотренных еще при инициализации.

Далее идет определение и выборка сервера с наименьшим количеством активных подключений S_i^{min} на коэффициент мощности сервера k_i и запрос перенаправляется на этот сервер.

Таки образом, модифицированный алгоритм Least Connections для кластеров «single-instance application» содержит преимущественную разницу с исходным образцом за счет разметки весов на серверах. Разработанный авторами программный код содержит выполнение сортировки узлов кластера при соответствии с количеством активных подключений и коэффициентом нагрузки на каждый отдельный узел, и выдачей соответственного адреса. Кроме того, модифицированный алгоритм Least Connections в силу своих динамических характеристик может распределять равномерно нагрузку по всем узлам серверов, развернутых в кластере «single-instance application».

Заключение

Проведенные исследования и реализация алгоритмов балансировки Round Robin, Weighted Round Robin, Load Least, Least Connections, тестирование полученных результатов приводят к следующим выводам:

- алгоритм Least Connections является достаточно эффективным для решения задачи балансировки нагрузки на серверах, развернутых в кластере «single-instance application»;
- получена равномерная балансировка нагрузки в узлах серверов;
- применение предложенных модификаций алгоритма дает возможность масштабирования приложений и повышает отказоустойчивость за счет равномерного распределения нагрузки по узлам, что так же повышает отказоустойчивость системы;

В целом, в результате проведенных исследований можно сделать заключение, что при использовании алгоритма Least Connections достигается уменьшение рисков сбоя «слабых» узлов сервера за счет определения вычислительных характеристик и введения коэффициента мощности сервера. Кроме того, данный алгоритм способствует уменьшению излишних задержек и в силу своих динамических характеристик может распределять равномерно нагрузку по всем узлам серверов (тип).

Таким образом, проведенные авторами данной статьи исследования позволили провести анализ и решение задачи балансировки нагрузки применением алгоритма Least Connections в соответствии с требованиями повышения эффективности и повышения производительности распределения нагрузки.

Список использованной литературы:

- 1 Joldasbayev S.K., Balakayeva G.T., Aidarov K.A., Chris Phillips Dynamic request distribution for enhanced Quality of Service // Вестник КазНУ, серия математика, механика, информатика, 4 (100), 2018, p 18-27.
- 2 Джолдасбаев С., Балакаева Г.Т., Айдаров К.А., Даркенбаев Д.К. Проектирование и исследование интеллектуальной системы предоставления услуг// Материалы XIV международной азиатской школы-семинара "Проблемы оптимизации сложных систем", Kyrgyzstan, 7/20/2018-7/31/2018
- 3 Liu, S., Ren, S., Quan, G., Zhao, M., Ren, S. Profit Aware Load Balancing for Distributed Cloud Data Centers. 2013 IEEE 27th International Symposium on Parallel and Distributed Processing, (2013): 611–622.
- 4 Karger D. R., Ruhl M. Simple efficient load balancing algorithms for peer-to-peer systems //Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures. – ACM, 2004. – с. 36-43.
- 5 Айдаров К., Балакаева Г. Исследование алгоритмов и методов балансировки нагрузки и построение моделей для сетей массового обслуживания, Марчуровские научные чтения - 2017, Труды международной научной конференции. 2017, Издательство: Институт вычислительной математики и математической геофизики Сибирского отделения РАН (Новосибирск).
- 6 Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Глава 11. Хеш-таблицы. //Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. - 2-е изд. -М.: Вильямс, 2005. - 1296 с. - ISBN 5-8459-0857-4.
- 7 Кочетов Ю. А., Кочетова Н. А. Задача балансировки нагрузки на серверы // Вестн. Новосиб. гос. ун-та. Серия: Информационные технологии. 2013. Т. 11, вып. 4. С. 71–76.
- 8 Lee, Y. C., Zomaya, A. Y. Energy efficient utilization of resources in cloud computing systems. The Journal of Supercomputing, 60(2), (2010): 268–280.
- 9 Enokido, T., Aikebaier, A., Takizawa, M. A Model for Reducing Power Consumption in Peer-to-Peer Systems. IEEE Systems Journal, 4(2),(2010): 221–229.
- 10 Mukherjee, M., Shu, L., Wang, D. Survey of Fog Computing: Fundamental, Network Applications, and Research Challenges. IEEE Communications Surveys and Tutorials, (2018): 1–1.
- 11 Nagpure, M. B., Dahiwale, P., Marbate, P. An efficient dynamic resource allocation strategy for VM environment in cloud. 2015 International Conference on Pervasive Computing (ICPC) (2015).
- 12 Vakiliinia, S., Heidarpour, B., Cheriet, M. Energy Efficient Resource Allocation in Cloud Computing Environments. IEEE Access, 4, (2016), 8544–8557.
- 13 Zhang, W., Zhang, Z., Chao, H.-C. . Cooperative Fog Computing for Dealing with Big Data in the Internet of Vehicles: Architecture and Hierarchical Resource Management. IEEE Communications Magazine, 55(12),(2017): 60–67.
- 14 Hameed, A., Khoshkbarforoushha A., Ranjan, R., Jayaraman, P. P., Kolodziej, J., Balaji, P., Zomaya, A. A survey and taxonomy on energy efficient resource allocation techniques for cloud computing systems. Computing, 98(7), (2014): 751–774.
- 15 Ge, Y., Zhang, Y., Qiu, Q., Lu, Y.-H. A game theoretic resource allocation for overall energy minimization in mobile cloud computing system. Proceedings of the 2012 ACM/IEEE International Symposium on Low Power Electronics and Design - ISLPED '12., (2012): 279-284.

МРНТИ 06.54.51:06.73.15
УДК 336.741.24

DOI: <https://doi.org/10.51889/2020-1.1728-7901.55>

Б.А. Досжанов

Қорқыт Ата атындағы Қызылорда мемлекеттік университеті, Қызылорда қ., Қазақстан

БЛОКЧЕЙН ТЕХНОЛОГИЯСЫ МЕН БИТКОЙН КРИПТОВАЛЮТАСЫ: ЖҰМЫС ҰСТАНЫМЫ ЖӘНЕ ЕРЕКШЕЛІКТЕРІ

Аңдатпа

Үкіметтік деңгейде қолға алынған әлеуметтік-экономикалық салалармен қатар мемлекеттік-жекеменшік әріптестіктерде, компаниялар мен мекемелердің, ұйымдардың қызметтерінде интернет-технологияларға айрықша мән берілуі блокчейндердің ірі инфрақұрылымға бірігуіне мүмкіндік беруде. Мұндай нәтижеге пайдаланушылар тарапынан қолжетімділікті қамтамасыз ету, мәліметтер базасын сақтаудың ішкі алгоритмдерін өзгерту арқылы жетуге болады. Блокчейн технологиясының басты ерекшелігі – ол жүйені орталықсыздандыруға негізделген. Кез-келген қорғаныс құралдарын пайдалану жағдайында серверде орналасқан мәліметтер базасына бұзып кіруге болатын болса, блокчейнде мұндай келеңсіздікке жол берілмейді.

Мақалада блокчейн технологиясы мен биткойн криптовалютасына түсініктеме беріледі және олардың жұмыс ұстанымы қарастырылады. Олардың болашақтағы перспективалық бағыттары мен интернет арқылы жүргізілетін қаржылық іс-әрекеттегі ауқымды ықпалы баяндалады. Сондай-ақ, криптовалюталар мен онда қолданылатын технологиялар, алгоритмдер сипатталады.

Түйін сөздер: криптовалюта, блокчейн технологиясы, биткойн, электрондық ақша айналымы.