Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н. Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н. Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

«Аналитический обзор алгоритмов консенсуса в распределенных системах»

Студент ИУ7-53Б _____ Д. В. Недолужко
(Группа) (Подпись, дата) (И. О. Фамилия)

Руководитель курсовой работы _____ Б. К. Аристов
(Подпись, дата) (И. О. Фамилия)

2022 г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ **3**

1 Анализ предметной области **4**

1.1 Подходы к организации многосерверных систем 4

1.2 Задача достижение консенсуса 5

1.2.1 Типы отказоустойчивости 5

1.2.2 Эксклюзивные и инклюзивные алгоритмы 6

1.3 Блокчейн 6

1.4 Вывод 7

2 Классификация существующих решений **8**

2.1 Существующие решения 8

2.1.1 Алгоритмы, основанные на голосовании 8

2.1.2 Алгоритмы, основанные на доказательстве 12

2.2 Критерии оценивания 13

2.2.1 Классификация по модели принятия решения 13

2.2.2 Классификация по эксклюзивности 14

2.2.3 Классификация по типу отказоустойчивости 15

2.3 Вывод 15

ЗАКЛЮЧЕНИЕ **17**

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ **18**

ВВЕДЕНИЕ

В современном мире все большую популярность приобретают распределенные системы. Переход к ним аргументируется лучшей масштабируемостью и отказоустойчивостью. Однако распределенные системы порождают новые задачи, связанные с согласованными принятиями решений между узлами системы. Принятие таких решений описывают алгоритмы достижения консенсуса.

Целью данной работы является обзор существующих алгоритмов консенсуса.

Для достижения поставленной цели требуется решить следующие задачи:

- определить основные термины, связанные алгоритмами консенсуса;
- рассмотреть существующие алгоритмы;
- выделить критерии классификации алгоритмов;
- провести классификацию алгоритмов.

1 Анализ предметной области

Приложения, работающие с большими объемами данных проникли во все сферы нашей жизни. Банковские системы, бронирование отелей, интернет магазинов — все они сталкиваются с задачами надежного хранения и обработки больших объемов данных.

1.1 Подходы к организации многосерверных систем

Существует 3 основных подхода к организации систем, состоящих из нескольких вычислительных машин[1]:

1. Централизованный
2. Децентрализованный
3. Распределенный

Наиболее простым в организации работы подходом является централизованный. При нем выделяется главный сервер, на который ложится ответственность за управление всем кластером. Зависимые сервера обмениваются сообщениями только с главным сервером и не общаются между собой. Такой подход порождает множество проблем: такими системы являются слабо масштабируемыми и обладают слабой отказоустойчивостью, ведь для приведения системы в неработоспособное состояние достаточно падения только одного главного узла.

Децентрализованный подход пытается решить проблемы централизованного подхода. При нем существуют несколько главных сервером, а так же зависимые от них. Каждых из зависимых серверов общается со своим главным сервером. Такая система является устойчивой к отказу в случае падения одного из главным серверов.

В распределенных системах все узлы системы являются равными, среди нет главных серверов. Каждый из узлов способен обрабатывать запросы. Такая система наиболее устойчива к падению и обладает наилучшей масштабируемостью.

Организация связей в данных подходах изображены на рисунке 1.1.

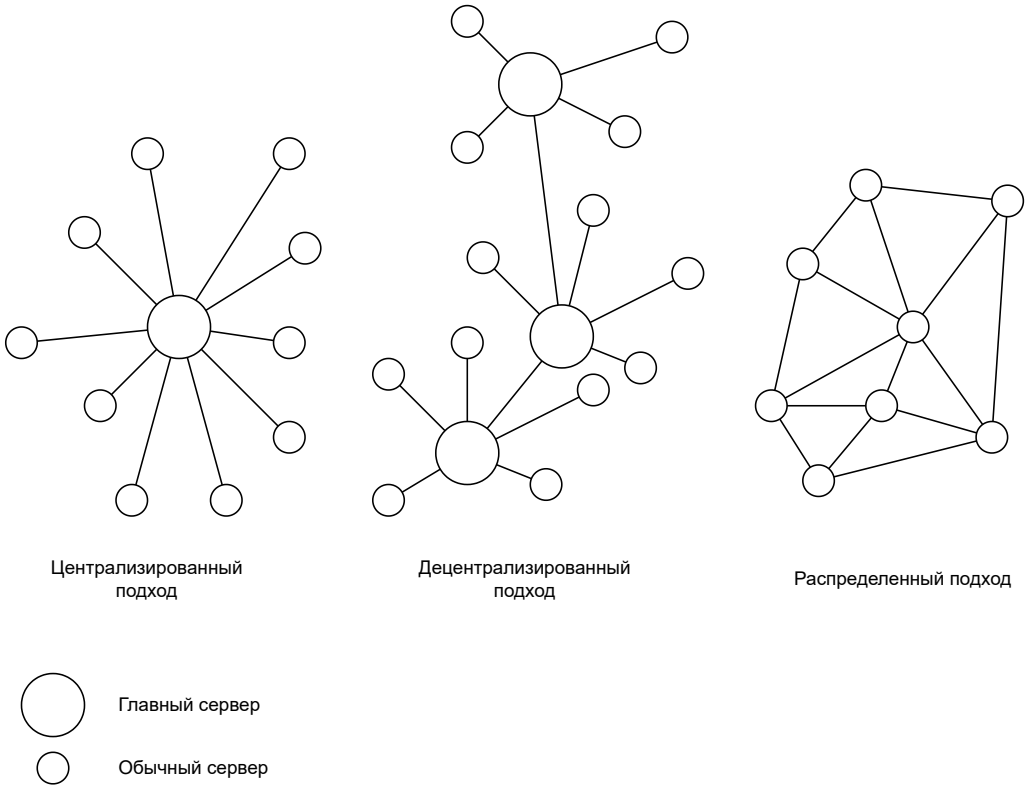

Рисунок 1.1 – Организация связей между серверами

1.2 Задача достижение консенсуса

Фундаментальной проблемой в распределенных системах является достижение общей надежности системы. Для ее достижения необходима координация процессов для достижения общего соглашения по поводу принятия или непринятия некоторого значения всей системой — задача консенсуса[2]. Примерами такой работы может являться соглашение по поводу некоторого единственного значения или задача репликации журнала[3].

1.2.1 Типы отказоустойчивости

В распределенных системах в работе участвует множество вычислительных машин, каждая из которых может выйти из строя. Рассматривают 2 типа алгоритмов достижения консенсуса по принципу отказоустойчивости:

- устойчивость к падению
- византийская отказоустойчивость

В первом случае рассматриваются сбои связанные с отказом оборудования, ошибки в программном обеспечении, сбои в сети. Алгоритмы устойчивые

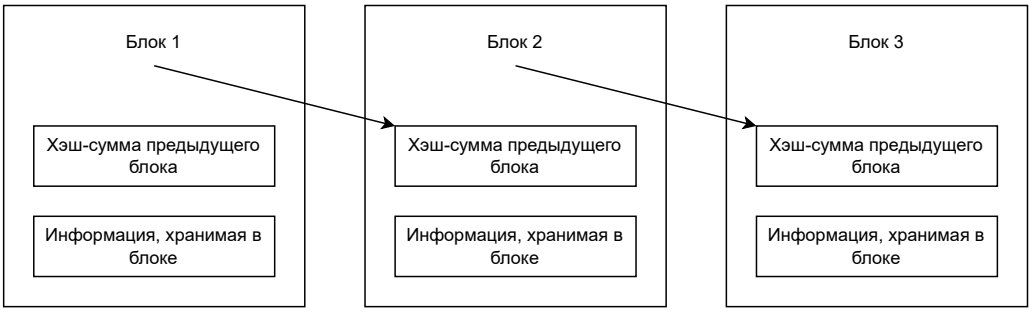

Рисунок 1.2 – Пример цепочки блокчейна

к падению не обрабатывают умышленные вредоносные действия в системе. Под византийской же устойчивостью подразумевается обработка в том числе и вредоносных действий узлов: посылка некорректных сообщений, посылка ложной информации, попытка вывести систему из согласованного состояния.

1.2.2 Эксклюзивные и инклюзивные алгоритмы

Алгоритмы достижения консенсуса классифицируются по модели обеспечения доступа к сети на следующие типы[4]:

- Эксклюзивные
- Инклюзивные

В эксклюзивных алгоритмах достижения консенсуса принимать участие в работе алгоритма могут только заранее установленные узлы в ограниченном количестве. В инклюзивных алгоритмах такое ограничение снимается, принимать участие в них может любой желающий узел.

1.3 Блокчейн

Важным толчком в развитии и разработке алгоритмов консенсуса послужило появление криптовалют, построенных поверх технологии блокчейна.

Блокчейн — выстроенная по определенным правилам непрерывная последовательная цепочка блоков (связный список), содержащих информацию. Связь между блоками обеспечивается не только нумерацией, но и тем, что каждый блок содержит свою собственную хеш-сумму и хеш-сумму предыдущего блока. Изменение любой информации в блоке изменит его хеш-сумму.

Пример цепочки блокчейна приведен на рисунке 1.2

1.4 Вывод

В данном разделе была обоснована актуальность поставленной задачи, определены основные термины, связанные с алгоритмами достижения консенсуса.

2 Классификация существующих решений

2.1 Существующие решения

Задача достижения консенсуса берет свое начало с публикации в 1982 году Лесли Лэмпортом, Робертом Шостаком и Маршалом Пизосом задачи о Византийских генералах[5]. В данной задаче рассматривается алгоритм принятия согласованного решения группой турецких генералов путем обмена сообщениями.

На текущий момент разработано множество алгоритмов консенсуса, каждый из которых решает собственный подкласс задач.

Одним из видов классификации алгоритмов консенсуса является классификация по принципу принятия решения [6]:

- алгоритмы основанные на голосовании
- алгоритмы основанные на доказательстве

2.1.1 Алгоритмы, основанные на голосовании

Общей чертой алгоритмов, основанных на голосовании является выдвижение нового значения одним из участников и последующее голосование за принятия данного значения. Если определенная доля участников голосуют за новое значение, то оно считается принятым

Паксос

Паксос[7] — алгоритм консенсуса, предназначенный для согласования одного единственного значения группой участников. Данный алгоритм является эксклюзивным, устойчивым к падению.

Алгоритм Паксос определяет 3 роли для процессов участников:

1. Заявитель
2. Избиратель
3. Ученик

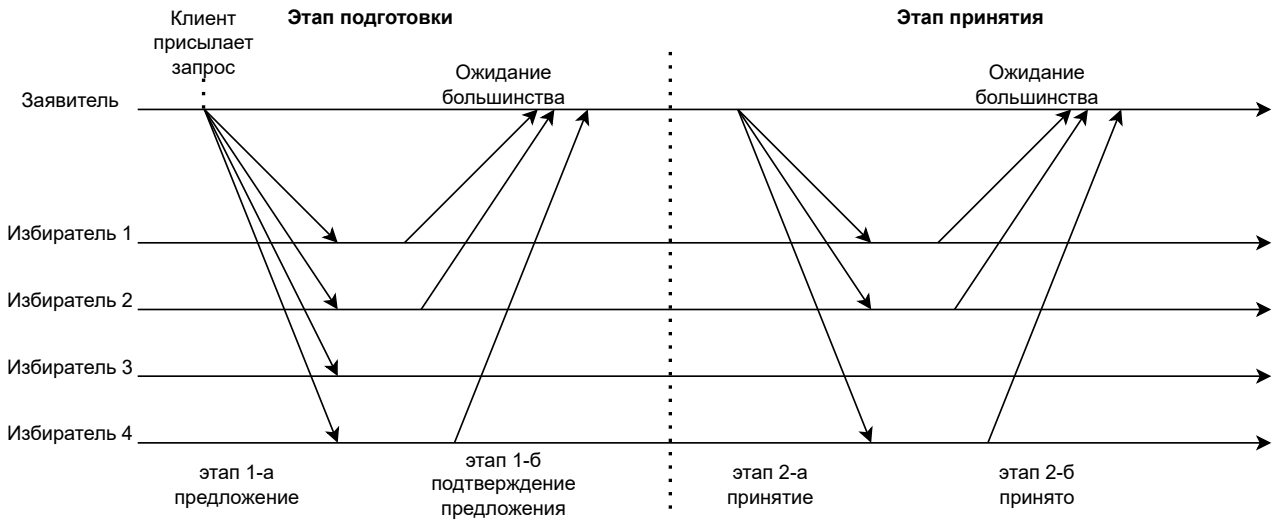

Рисунок 2.1 – Пример работы алгоритма Паксос

Каждый из узлов может принимать несколько из ролей одновременно. Заявитель получается новое значение от клиента, выдвигает его избирателям и предлагает проголосовать за него. Избиратель ответственен за голосование по предложению заявителя. Ученик информируется о результатах голосования, но не принимает в нем участия.

Выполнение алгоритма разделяется на 2 этапа: подготовка и принятия. Пример работы алгоритма представлен на рисунке 2.1.

Этапы выполнения:

1. Подготовка. На данном этапе заявитель рассылает избирателям сообщения с номером раунда и новым значения для голосования. Избиратели голосуют, готовы ли они принять данное значение. Этап подготовки необходим для определения, является ли текущее предложение актуальным. Актуальность предложения определяется по номеру раунда. Избиратели сохраняют последний номер раунда, за который они голосовали. Если в предложении номер раунда выше, чем сохраненный номер, то предложение считается актуальным и избиратель сообщает заявителю о свой готовности на него проголосовать.

2. Принятие. На данном этапе заявитель определяет сколько избирателей готовы проголосовать за новое значение. Если готовы проголосовать более половины избирателей, то рассылается сообщение о фиксации данного значения и значение считается принятым системой.

Рафт

Рафт[8] — алгоритм консенсуса, предназначенный для решения задачи репликации журнала. Данный алгоритм является эксклюзивным, устойчивым к падению.

Алгоритм Рафт основывается на идеях алгоритма Паксос, но решает проблему его низкой производительности. Так в алгоритме Паксос для определения каждым узлом актуального значения необходимо обменяется n^2 сообщениями, где n — число узлов в сети, так как каждый узел должен опросить все остальные узлу о принятом ими значении. Рафт решает данную проблему введением явного лидера, ответственного за информирование участников об актуальном состоянии принятого значения.

Алгоритм Рафт определяет 3 роли:

1. Лидер
2. Последователь
3. Кандидат

Во всей системе может быть только один лидер, он ответственен за получение сообщений от клиента, управление журналом и общение с последователями. Последователь ответственен за сохранение записей журнала, получаемых от лидера. Кандидатом становится последователь, в случае долгого не получения сообщений от лидера. Кандидат выдвигает свою кандидатуру в качестве лидера и проводит голосование. Диаграмма переходов ролей представлена на рисунке 2.2.

Алгоритм Рафт вводит понятие эры. Эра — период работы лидера. Концом и началом каждой эры является выбор нового кандидата в качестве лидера.

pBFT

pBFT[9] — алгоритм консенсуса, предназначенный для согласования единственного значения в сети, где возможно византийское поведение участников. Данный алгоритм является инклюзивным.

Данный алгоритм определяет 2 роли:

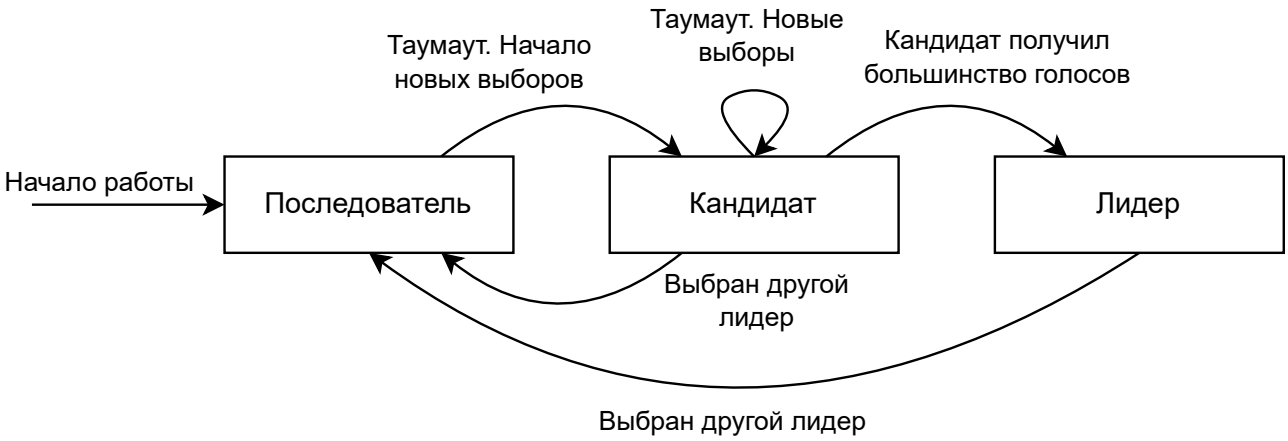

Рисунок 2.2 – Диаграмма переходов ролей в алгоритме Рафт

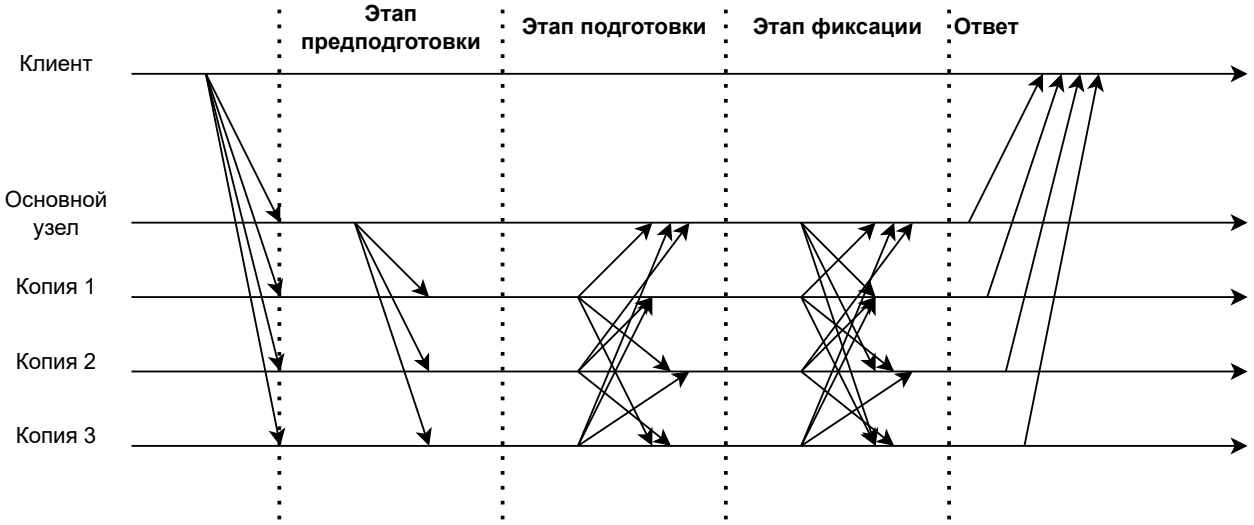

Рисунок 2.3 – Пример работы алгоритма pBFT

1. Основной узел
2. Копия

Основной узел инициирует начало работы алгоритма, в дальнейшем обменимается сообщениями с копиями на равных условиях.

pBFT определяет 3 этапа: предподготовка, подготовка, фиксация. Пример работы алгоритм представлен на рисунке 2.3.

Этапы выполнения:

1. В самом начале клиент рассылает сообщение на все узлы системы.
2. Предподготовка. На данном этапе основной узел рассылает следующие сообщения на копии $<PRE\text{-}PREPARE,\ H(m),\ s,\ v>$, где $H(m)$ — хэш-

сумма сообщения клиента, s — номер сообщения, v — последовательный номер основного узла. Данное сообщение подписывается электронном подписью перед отправкой.

3. Подготовка. При получении копией сообщения, копия проверяет электронную подпись, хэш-сумму сообщения и номер основного узла. В случае успешной проверки копия посылает следующее сообщения на все узлы сети: *<PREPARE, H(m), s, v>*, где *H(m)*

4. Фиксация. Если узел получил f некорректных сообщений и не меньше чем $2f + 1$ корректных сообщений, то копия устанавливает значение и отправляет подтверждение клиенту. Значение считается установленным.

Алгоритм pBFT корректно работает в случае $n >= 3f+1$, где n — число честных узлов, f — число византийских узлов.

2.1.2 Алгоритмы, основанные на доказательстве

Алгоритмы, основанные на доказательстве, предназначены для работы в сетях с неограниченным количеством узлов, поэтому в них не может быть применено голосование, так как злоумышленник может владеть неограниченным количеством узлов. Альтернативой подходу голосования является доказательство узлам сети своего более квалифицированного права на фиксацию значения.

Доказательство выполнения работы

Доказательство выполнения работы[10] — алгоритм консенсуса, применяемый в блокчейн проектах, требующий от участников сети решение сложного криптографического пазла. Решение пазла не может быть предсказано заранее и требует от узла долгих вычислений. Блок узла, первого решившего задачу, добавляется в блокчейн. Примером такой задачи является генерация от блока хэш-суммы, меньшей заданного значения. Схема работы алгоритма представлена на рисунке 2.4

Данный алгоритм считается устойчивым к византийскому поведению. Он сохраняет свою работоспособность, если злоумышленник владеет не больше чем 50% вычислительных мощностей.

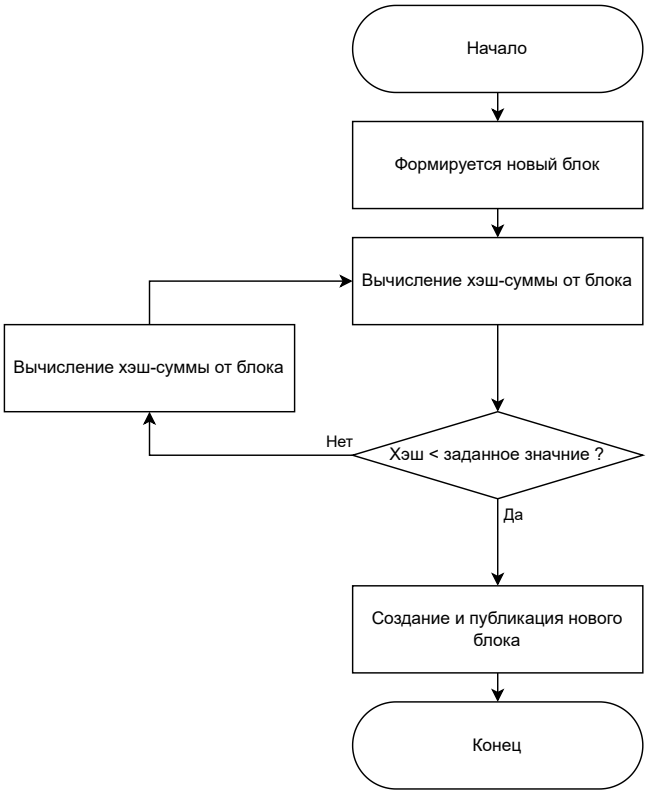

Рисунок 2.4 – Схема алгоритма доказательства выполнения работы

Доказательство доли владения

Доказательство доли владения[11] — еще алгоритм консенсуса, применяемый в блокчейн проектах. Данный алгоритм является альтернативой алгоритму доказательству выполнения работы и решает проблему больших вычислительных затрат для добавления нового блока в блокчейн.

При использовании этого метода алгоритм формирования блока не зависит от мощности оборудования, но с большей вероятностью блок будет сформирован той учетной записью, у которой текущий баланс больше. Например, участник, владеющий 1% от суммарного количества, в среднем будет генерировать 1% новых блоков.

2.2 Критерии оценивания

2.2.1 Классификация по модели принятия решения

По модели принятия решений алгоритмы консенсуса делятся на 2 типа:

1. Алгоритмы, основанные на голосовании
2. Алгоритмы, основанные на доказательстве

Общей чертой алгоритмов, основанных на голосовании, является выдвижение нового значения и голосовании на него участниками системы. Значение считается принятым, если за него проголосовали больше половины участников. Такие алгоритмы применяется в эксклюзивных сетях, где устанавливается количество участников, что позволяется определить отношение количества проголосовавших участников ко всем участникам.

В алгоритмах, основанных на доказательстве, чаще применяются в инклюзивных блокчейнах. Участники системы соревнуются между собой за право создать новый блок.

Классификация алгоритмов по данному критерию представлена в таблице 2.1.

Таблица 2.1 – Классификация алгоритмов консенсуса по методу принятия решения

Алгоритм	**Тип**
Паксос	Основанный на голосовании
Рафт	Основанный на голосовании
pBFT	Основанный на голосовании
Доказательство выполнения работы	Основанный на доказательстве
Доказательство доли владения	Основанный на доказательстве

2.2.2 Классификация по эксклюзивности

По типу эксклюзивности алгоритмы консенсуса делятся на 2 типа:

1. Экслюзивные
2. Инклюзивные

В эксклюзивных алгоритмах достижения консенсуса принимать участие в работе алгоритма могут только заранее установленные узлы в ограниченном количестве. В инклюзивных алгоритмах такое ограничение снимается, принимать участие в них может любой желающий узел

Классификация алгоритмов по данному критерию представлена в таблице 2.2.

Таблица 2.2 – Классификация алгоритмов консенсуса по эксклюзивности

Алгоритм	**Тип**
Паксос	Эксклюзивный
Рафт	Эксклюзивный
pBFT	Эксклюзивный
Доказательство выполнения работы	Инклюзивный
Доказательство доли владения	Инклюзивный

2.2.3 Классификация по типу отказоустойчивости

По типу отказоустойчивости алгоритмы консенсуса делятся на 2 типа:

1. Устойчивость к падению
2. Византийская отказоустойчивость

В первом случае рассматриваются сбои связанные с отказом оборудования, ошибки в программном обеспечении, сбои в сети. Алгоритмы устойчивые к падению не обрабатывают умышленные вредоносные действия в системе. Под византийской же устойчивостью подразумевается обработка в том числе и вредоносных действий узлов: посылка некорректных сообщений, посылка ложной информации, попытка вывести систему из согласованного состояния.

Степень отказоустойчивости алгоритма алгоритма определяется в виде отношения числа честных узлов к числу неработающих или византийских узлов. Данное отношение задается в форме $n \geq af + b$, где n — число честных узлов, f — число некорректных узлов, a, b— некоторые действительных числа.

Классификация алгоритмов по данному критерию представлена в таблице 2.3.

2.3 Вывод

В данном разделе были выделены основные критерии классификации алгоритмов, и проведена классификация по ним. Выделены следующие критерии: модель принятия решения, эксклюзивность, тип отказоустойчивости.

Каждый из приведенных алгоритмов используется в зависимости от поставленных целей. Так алгоритм Рафт используется в закрытых распределенных системах, где участники заранее определены и доверяют друг другу.

Таблица 2.3 – Классификация алгоритмов консенсуса по типу и степени отказоустойчивости

Алгоритм	**Тип отказоустойчивости**	**Степень отказоустойчивости**
Паксос	Устойчивость к падению	$n \geq 2f + 1$
Рафт	Устойчивость к падению	$n \geq 2f + 1$
pBFT	Византийская отказоустойчивость	$n \geq 3f + 1$
Доказательство выполнения работы	Византийская отказоустойчивость	$n \geq f + 1$
Доказательство доли владения	Византийская отказоустойчивость	$n \geq f + 1$

Алгоритм pBFT применяется в случае ожидаемого византийского поведения участников. А алгоритмы доказательства работы и доказательства доли владения используются в инклюзивных блокчейн проектах для подтверждения блоков.

ЗАКЛЮЧЕНИЕ

В ходе научно-исследовательской работы были рассмотрены основные алгоритмы достижения консенсуса. Можно сделать вывод, что не существует универсальных алгоритмов консенсуса, каждых из них используется в зависимости от поставленных целей. При выборе алгоритма стоит учитывать является ли сеть открытой к новым участникам, возможно ли византийское поведение участников.

Так же были выполнены следующие задачи:

- определены основных терминов, связанных с алгоритмами консенсуса;
- рассмотрены существующих алгоритмов;
- выделены критериев классификации алгоритмов;
- проведена классификацию алгоритмов.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Why distributed?: A critical review of the tradeoffs between centralized and decentralized resources / S. P. Burger [и др.] // IEEE Power and Energy Magazine. — 2019. — Т. 17, № 2. — С. 16—24.
2. Consensus in distributed soft environments / C. Carlsson [и др.] // European Journal of Operational Research. — 1992. — Т. 61, № 1/2. — С. 165—185.
3. *Panda S. K.*, *Naik S.* An efficient data replication algorithm for distributed systems // International Journal of Cloud Applications and Computing (IJCAC). — 2018. — Т. 8, № 3. — С. 60—77.
4. *Butun I.*, *Österberg P.* A review of distributed access control for blockchain systems towards securing the internet of things // IEEE Access. — 2020. — Т. 9. — С. 5428—5441.
5. *Lamport L.* The weak Byzantine generals problem // Journal of the ACM (JACM). — 1983. — Т. 30, № 3. — С. 668—676.
6. *Nguyen G.-T.*, *Kim K.* A survey about consensus algorithms used in blockchain // Journal of Information processing systems. — 2018. — Т. 14, № 1. — С. 101—128.
7. *Lamport L.* Paxos made simple // ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001). — 2001. — С. 51—58.
8. *Ongaro D.*, *Ousterhout J.* In search of an understandable consensus algorithm // 2014 USENIX Annual Technical Conference (Usenix ATC 14). — 2014. — С. 305—319.
9. Practical byzantine fault tolerance / M. Castro, B. Liskov [и др.] // OsDI. Т. 99. — 1999. — С. 173—186.
10. On the security and performance of proof of work blockchains / A. Gervais [и др.] // Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. — 2016. — С. 3—16.
11. *King S.*, *Nadal S.* Ppcoin: Peer-to-peer crypto-currency with proof-of-stake // self-published paper, August. — 2012. — Т. 19, № 1.