

## Методы и способы анализа надежности взаимодействия распределенных систем

О.С. Рудько, В.А. Рубин, Н.А. АЛЕШИН

Дано краткое описание распределенных систем и их надежности. Приводится обоснование преимуществ распределенных систем. Приведены формулы нахождения надежности и основные способы анализа распределенных систем. Рассматриваются примеры применения алгоритмов консенсуса.

**Ключевые слова:** надежность, отказоустойчивость, распределенные системы, консенсус.

A brief definition of distributed systems and reliability is presented. The substantiation of the advantages of distributed systems is considered. The formulas for finding the reliability and the main methods for the analysis of distributed systems are given. Examples of application of consensus algorithms are studied.

**Keywords:** reliability, fault tolerance, distributed systems, consensus.

**Введение.** Состояние современного общества неразрывно связано с процессами информатизации, затрагивающими практически все сферы человеческой деятельности. За периодом местной компьютеризации появилась эра создания распределенных информационных систем [1]–[3]. Распределенная система – это набор независимых компьютеров, объединенных в единое целое для представления пользователям. Важной особенностью распределенной системы является то, что от человека скрыто каким образом происходит процесс общения компонентов системы. Распределенные системы довольно легко поддаются масштабированию, расширению и тестированию. Эта характеристика является прямым следствием независимости и индивидуальности ее компонентов. Распределенные системы обычно работают безотказно, но некоторые их части могут временно выходить из строя. Пользователи не должны знать о том, что эти части подверглись изменению, или что добавлены новые части для поддержания работы распределенной системы.

Надежность распределенной системы является важной характеристикой, определяющей вероятность того, что программная система будет работать по спецификации. Ненадежность компонентов, осуществляющих передачу информации, может заключаться в полном прекращении передачи, в одностороннем прекращении передачи, в возникновении случайных ошибок или помех при передаче.

Для того, чтобы получить представление различных компьютеров и сетей в виде единой системы, разработка распределенных систем часто включает в себя особый уровень программного обеспечения, который находится между самым верхним уровнем, на котором присутствуют пользователи, и нижним уровнем, который состоит из низкоуровневых операционных систем (рисунок 1).



Рисунок 1 – Распределенная система организована в виде службы промежуточного уровня

Классическим примером распределенной системы является World Wide Web. Всемирная паутина представляет единообразную, простую и целостную модель распределенных документов. Для того, чтобы увидеть документ, пользователю необходимо активизировать ссылку, после чего документ отобразится на экране. Отпадает необходимость знать, с какого сервера доставляется документ, достаточно лишь информации о том, где он расположен. Публикация документа тоже является достаточно простой операцией, требуется задать ему уникальное имя в форме унифицированного указателя ресурса, которое ссылается на локальный файл с содержимым документа.

**Основные способы анализа надежности распределенных систем.** Распределенная система состоит из модулей системы, которые в свою очередь состоят из компонентов.

Компоненты могут иметь:

1. Разную надежность.
2. Разную зависимость между компонентами.

Надежность системы – это функция надежности компонентов и зависимостей между компонентами.

Надежность последовательных систем, состоящих из множества компонентов, соединенных друг с другом, определяется вероятностью отказа каждого компонента. Отказ каждого компонента приводит к отказу всей системы. Надежность системы может быть выражена через надежность ее компонентов, если они отказывают независимо. Для этого используют следующую формулу:

$$R = \prod_{k=1}^{Q_p} R_k,$$

где  $Q_p$  – число компонентов,  $R_k$  – надежность  $k$ -го компонента, которые должны быть заданы в одном интервале.

Используя связь между надежностью и частотой ошибок, что выражается формулой:

$$\lambda = \frac{-\ln R}{\tau} \leftrightarrow R = e^{-\lambda\tau},$$

получаем что:

$$\lambda = \sum_{k=1}^{Q_p} \lambda_k,$$

где  $\lambda_k$  – частота ошибок  $k$ -го компонента, из чего можно сделать вывод, что общая частота отказа системы есть сумма частот ошибок отдельных ее компонентов. Отсюда следует вывод, что при последовательной связи компонентов распределенной системы вероятность безотказной работы уменьшается.

Надежность параллельных распределенных систем, состоящих из  $n$  компонентов, работающих параллельно, определяется вероятностью отказа всех компонентов одновременно. В таком случае вероятность отказа системы можно найти по формуле:

$$F(t) = 1 - R(t),$$

где  $F$  – вероятность отказа системы,  $R$  – вероятность безотказной работы.

Тогда вероятность надежности параллельной распределенной системы можно найти по формуле:

$$R(t) = 1 - \prod_{i=1}^n F_i(t) = 1 - \prod_{i=1}^n (1 - R_i(t)),$$

где  $F_i$  – вероятность отказа компонента параллельной системы,  $R_i$  – вероятность безотказной работы компонента системы.

Из приведенных выше формул можно сделать вывод, что надежность системы с параллельными компонентами значительно выше надежности отдельного компонента. Параллельное соединение – метод создания надежной распределенной системы из ненадежных компонентов.

**Алгоритмы повышения надежности.** На сегодняшний день существует несколько весомых алгоритмов, способных решать проблемы надежности [4]. Задача принятия консенсуса является одной из основополагающих для современных распределенных систем. Задача достижения консенсуса – это задача получения согласованного значения группой участников, члены которой называются узлами, в ситуации, когда возможны отказы отдельных участников, предоставление им некорректной информации и искажение переданных значений средой передачи данных.

Случай нетипичного функционирования компонентов распределенной системы можно охарактеризовать полным отказом компонента, который приводит к полной его недоступности, и византийской ошибкой – компонент продолжает работать, но при этом возвращает неверную информацию. Например, использование оперативной памяти без функции коррекции ошибок может привести к получению фиктивных данных из памяти, а ошибки оборудования могут приводить к изменению или повреждению пакетов данных.

Консенсус должен удовлетворять следующим условиям:

1. Согласованность – все узлы принимают одно значение.
2. Корректность – значение из тех, что предложено.
3. Конечность – каждый узел принимает решение за конечное число шагов.

Простейший способ реализовать распределенную систему – создать набор клиентов, которые посылают команды центральному серверу. Сервер должен быть описан как детерминированный конечный автомат, который выполняет в определенном порядке команды от клиентов.

Примером такой системы является банковская, где клиентами могут быть клерки, а состояние центрального узла может состоять из счетов и балансов пользователей банка. Снятие наличных могло бы быть выполнено через выполнение конечным автоматом команды, которая уменьшала бы баланс определенного счета, когда баланс был бы больше суммы снятия, а затем генерировала бы старый и новый баланс в качестве результата обработки команды.

Реализация, которая использует центральный сервер, не является отказоустойчивой, так как в случае отказа система перестанет обрабатывать команды клиентов. Для обеспечения отказоустойчивости системе следует использовать набор серверов, каждый из которых представляет собой конечный автомат. Клиент, для которого выполняется команда, может использовать любой сервер.

Распределенные автоматы реализуются с помощью реплицированного журнала. На каждом сервере хранится журнал, содержащий последовательность команд, которые его собственный конечный автомат выполняет в постоянном порядке.

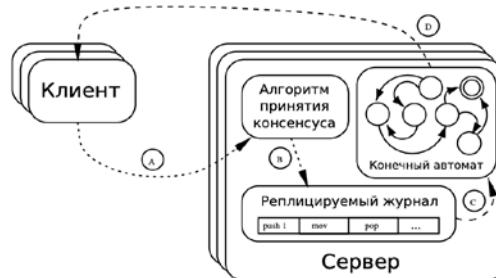


Рисунок 2 – Архитектура системы с использованием распределенного конечного автомата

Алгоритмы принятия консенсуса при практическом применении обладают следующими свойствами:

1. Обеспечивают безопасность при всех условиях, в том числе при задержках, потерях сообщений, их дублировании.
2. Высокая доступность серверов.
3. Команда считается выполненной только после подтверждения успешного ее выполнения. Меньшинство медленных узлов не должно повлиять на общую производительность системы.

**Заключение.** За последнее десятилетие распределенные системы набирают все большую популярность. В первую очередь это связано с тем, что процесс разработки таких систем намного быстрее монолитных в силу того, что компоненты систем могут разрабатываться независимо друг от друга разными людьми и даже командами, достаточно лишь того, чтобы соблюдался единый протокол общения между компонентами. Независимые компоненты системы просто тестировать и переиспользовать в разных частях системы. Важную роль играет возможность масштабируемости и отказоустойчивости. Современные алгоритмы принятия консенсуса позволяют создавать надежные распределенные системы, которые могут использоваться в банковских, аэрокосмических и других сложных отраслях, в которых критически важно соблюдение безотказной работы.

### Литература

1. Котляров, В. П. Проектирование отказоустойчивых распределенных информационных систем / В. П. Котляров. – СПб. : ОРЕЛЛИ, 2019. – 210 с.
2. Силен, Д. Основы Data Science и Big Data. Python и наука о данных / Д. Силен. – СПб. : ОРЕЛЛИ, 2017. – 336 с.
3. Бринк, Х. Машинное обучение / Х. Бринк. – СПб. : ДМК, 2017. – 336 с.
4. Клеппман, М. Высоконагруженные приложения. Программирование, масштабирование, поддержка / М. Клеппман. – СПб. : ОРЕЛЛИ, 2018. – 640 с.

Гомельский государственный  
университет им. Ф. Скорины

Поступила в редакцию 09.04.2021