

Увеличение степени отказоустойчивости в программно-аппаратных системах сетевого управления на примере мягкого облачного хранилища

Рассматриваются методы повышения отказоустойчивости сетевых систем управления на примере системы хранения данных типа мягкое облачное хранилище. На основе моделей прогнозирующего управления предлагается новая модель системы управления, способная противостоять таким негативным сетевым эффектам, как временные задержки и потеря пакетов. Разработан инструмент для тестирования новой модели системы управления методом внесения неисправностей.

Ключевые слова: отказоустойчивость, сетевые системы, мягкое облачное хранилище, программно-аппаратный

ВВЕДЕНИЕ

С развитием компьютерных сетей новым видом систем управления стали системы сетевого управления. Большой интерес к развитию и исследованию систем данного класса обусловлен тем, что во всем мире цифровые сети географически покрывают всё большую территорию и становятся всё более доступными, поэтому они могут применяться для реализации систем управления с обратной связью без дополнительных затрат на установку сетевого оборудования. Сетевые системы управления нашли свое применение в тех областях деятельности человека, где необходимо обеспечение критического уровня безопасности, надежности и отказоустойчивости, например: авиация, программная инженерия, космические аппараты, робототехника, химические и ядерные процессы. Ошибки в управлении такими сложными системами могут привести к крупным техногенным и экономическим катастрофам.

Ввиду большой актуальности и практической значимости изучения способов построения отказоустойчивых систем и их управления было опубликовано множество трудов как российских ученых: В.А. Богатырев, А.Е. Шумский, А.Н. Жирабок, А.В. Котенок, А.В. Ушаков, так и иностранных исследователей: М. Бланк, Ц. Чжао, Дж.Ч. Лапри, Е. Дуброва, Я. Лунц.

Объектом нашего исследования является система хранения данных (СХД) типа «мягкое облачное хранилище» (МОХ), а **предметом исследования** – её сетевая система управления. **Цель исследования** – повысить степень отказоустойчивости сетевой системы управления СХД «МОХ» при негативных сетевых эффектах таких, как сетевые задержки и потеря пакетов.

ОПИСАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ И ПОДХОДОВ К ОБЕСПЕЧЕНИЮ ЕЁ ОТКАЗОУСТОЙЧИВОСТИ

Рассмотрим структуру разрабатываемой нами системы, включающей объект управления и систему управления, основные причины возникновения отказов, а также используемый подход к обеспечению отказоустойчивости.

Отказоустойчивость. Использование компьютерного управления значительно упрощает применение сложных и эффективных алгоритмов управления. Отказоустойчивость становится одной из важнейших проблем при использовании этих алгоритмов [1, 2].

Отказоустойчивость – это способность управляемой системы выполнять цели управления, несмотря на возникновение отказов. При этом возможно ухудшение качества управления, что может быть допустимо. Отказоустойчивость достигается либо путем устранения отказов, либо через реконфигурацию системы или системы управления [3-5].

Одним из первых подходов к повышению отказоустойчивости компьютерной системы стало применение принципа избыточности, предложенного Джоном фон Нейманом [6] в 1950-х гг., который и сегодня широко используется при разработке различных технических систем. Наиболее эффективным методом является аппаратная избыточность, достигаемая путём резервирования аппаратных ресурсов системы. В ряде приложений отказоустойчивость путём резервирования является обязательным требованием, предъявляемым государственными надзорными органами к техническим системам.

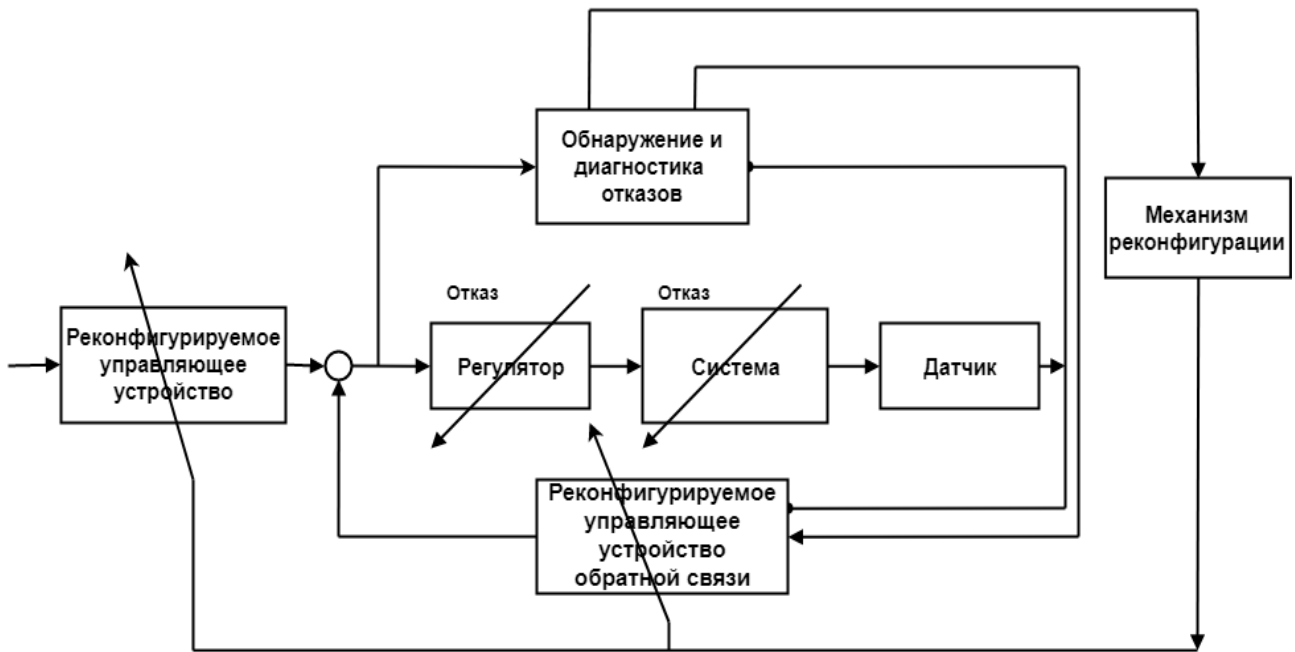


Рис. 1. Активная отказоустойчивая система

Выделим два основных типа отказоустойчивых систем – пассивные и активные (в зависимости от того, как в них реализуется избыточность).

Активные отказоустойчивые системы. В отличие от пассивных, активные отказоустойчивые системы (рис. 1) реагируют на неисправности компонентов, выстраивая управляющие воздействия таким образом, чтобы можно было поддерживать производительность, стабильность и надежность управляемой системы на заданном уровне. Этот подход основан на обнаружении и идентификации неисправностей в режиме реального времени для получения текущего состояния системы [7]. В дальнейшем эта информация используется системой управления для выполнения необходимой реконфигурации.

Временные ограничения обнаружения и идентификации неисправностей компонентов и реконфигурации системы являются критически важными критериями для активной отказоустойчивости.

Объект управления. В настоящей работе объектом управления является система хранения данных типа мягкое облачное хранилище (МОХ).

МОХ – это программно-определяемая облачная платформа для хранения данных, Любая организация, используя МОХ, может создать масштабируемые публичные облачные сервисы с надежностью и контролем инфраструктуры частного облака. Эта система хранения данных обеспечивает всестороннюю поддержку протоколов для неструктурированных данных и имеет гибкую программно-определяемую архитектуру, обеспечивающую неограниченную масштабируемость.

Графическое представление этих компонентов показывает рис. 2.



Рис. 2. Архитектура мягкого облачного хранилища

Мягкое облачное хранилище имеет специальные характеристики, которые необходимо поддерживать при решении задач управления, и как для любых других систем хранения данных, для этого хранилища характерен ряд контрольных показателей:

- вместимость – общий объем хранимой информации, которую может хранить система хранения данных. Вместимость выражается в единицах измерения количества информации, например, 256 терабайтов;

- задержка – время, необходимое для доступа к определенным данным, размещённым в системе хранения данных. Соответствующая единица измерения обычно составляет наносекунду для первичного хранения, миллисекунду – для вторичного хранения и секунду – для третичного хранения. Разделяют задержку чтения и задержку записи;

- пропускная способность – скорость, с которой информация может быть прочитана или записана в хранилище. В хранилищах данных пропускная способность обычно выражается в мегабайтах в секунду, но для некоторых случаев используют количество операций ввода/вывода в секунду. Это время, затрачиваемое системой хранения на выполнение операции ввода-вывода в секунду от начала до конца, измеряется в ОП/с. Как и в случае с задержкой, скорость чтения и скорость записи необходимо различать.

Система прогностического управления на базе внутренней модели объекта. Рассмотрим основные особенности систем прогностического управления (СПУ) для дискретных и непрерывных систем.

Непрерывная система может быть описана обыкновенным дифференциальным уравнением:

$$\dot{x}(t) = f(x(t), u(t)), \quad (1)$$

где $x(0) = x_0$, $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. Решением уравнения (1) является $x(t) = x(t, x_0, u)$ в момент времени t , вытекающее из функции управления $u: \mathbb{R} \rightarrow \mathbb{R}^m$ и удовлетворяющее начальным условиям. Множества возможных значений переменных состояния и управления обозначаются $X \subseteq \mathbb{R}^n$ и $U \subseteq \mathbb{R}^m$, соответственно. Если условия:

$$x(t, x_0, u) \in X, t \in [0, T] \quad \text{и} \quad u(t) \in [0, T] \quad (2)$$

выполнены, то функция управления u называется достижимой на интервале $[0, T]$ для x_0 , и обозначается как $u \in U(x_0, T)$. Здесь мы предполагаем существование и уникальность решения $x(\cdot, x_0, u)$ на интервале $[0, T]$ для всех $u \in U(x_0, T)$. Кроме того, управление u является достижимым для всех моментов времени на интервале $[0, T]$ относительно x_0 , т.е. $u \in U(x_0, \infty)$, если $u \in U(x_0, T)$ выполняется для любого $T > 0$.

Дискретная система может быть получена из непрерывной путём её дискретизации по времени. Например, дискретизация может быть реализована с помощью экстраполятора нулевого порядка. Если обозначить $\delta \in (0, T]$, $X \in \mathbb{R}^n$ и вектор управления $u \in \mathbb{R}^m$, то дискретную систему можно определить как:

$$\begin{aligned} x^+ &= f(x, u) = x(\delta, x, \tilde{u}), \\ \tilde{u}(t) &= u, t \in [0, \delta), \end{aligned} \quad (3)$$

где x^+ является решением дифференциального уравнения (1), полученного для $u = \text{const}$ на интервале времени δ , длящемся начиная с текущего состояния x . Тогда дискретное время k соответствует непрерывному времени $k\delta$. Таким образом, мы можем взаимодействовать с системой с непрерывным временем в режиме дискретного времени, т.е. мы можем описать дискретную систему с помощью разностного уравнения:

$$x(k+1) = f(x(k), u(k)), \quad (4)$$

где $x(0) = x_0$, $f: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$. Решение уравнения для $(u(k))_{k \in \mathbb{N}_0}$ и начального условия x_0 в момент времени k обозначается $x_u(k, x_0)$.

Входная последовательность

$$u = (u(0), u(1), \dots, u(N-1))$$

длины N называется допустимой для $x_0 \in X$, если выполнены условия:

$$f(x_u(k, x_0), u(k), u(k)) \in X. \quad (5)$$

Это справедливо для каждого $k \in \{0, 1, \dots, N-1\}$, для которого $u \in U(x_0, N)$. Аналогичное справедливо и для непрерывных систем, где входная последовательность $(u(k))_{k \in \mathbb{N}_0}$ называется допустимой для всех моментов времени относительно x_0 , т.е. $u \in U(x_0, \infty)$, если $u \in U(x_0, N)$ выполняется для каждого $N \in \mathbb{N}$.

Рассмотрим систему с дискретным временем (4). Наша цель – обеспечить устойчивость стационарного состояния x^* системы (4) с помощью $u: \mathbb{R}^n \rightarrow \mathbb{R}^m$, минимизируя целевую функцию:

$$J_\infty(x_0, u) = \sum_{k=0}^{\infty} \ell(x_u(k, x_0), u(k)), \quad (6)$$

так, чтобы состояние и ограничения управления были удовлетворительными, т.е. $u \in U(x_0, \infty)$. Целевая функция одного шага $\ell: \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \geq 0$ должна быть построена в соответствии с ожидаемыми результатами и быть положительно определена по x .

Компенсация сетевых задержек и потери пакетов. В сетевых системах управления измерительная информация, полученная датчиками, и входной сигнал, вычисленный системой управления, передается через сеть связи. Как следствие, могут возникать потери пакетов и сетевые задержки.

В этой ситуации система прогностического управления может использоваться для компенсации как задержек, так и потерь пакетов, например, передавая не только первое значение вектора управляющей последовательности u , но и большие части этой последовательности. Это поможет создать резервные входные воздействия в случае сбоя следующей передачи из-за потери пакета.

Поскольку модели прогнозирующего управления предоставляют целую последовательность входных значений, необходимая последовательность легко доступна без каких-либо дополнительных вычислительных операций. Так же, поскольку система прогностического управления является методологией управления, основанной на прогнозировании, задержки в сети можно компенсировать с помощью легкодоступного алгоритма прогнозирования.

ПОСТАНОВКА ЗАДАЧИ ИССЛЕДОВАНИЯ И РАЗРАБОТКА МЕТОДОВ ЕЁ РЕШЕНИЯ

Рассмотрим дискретную систему (4), при $x(0) = x_0$.

Для формализации идеи передачи и применения управляющих последовательностей вместо отдельных значений, мы используем концепцию многошаговых законов управления с обратной связью.

Многошаговый закон управления с обратной связью может быть описан математически в виде $u_{m^*} : \mathbb{R}^n \times \{0, 1, \dots, m^* - 1\} \rightarrow \mathbb{R}^m$, ($m^* \in \mathbb{N}$) и генерирует допустимую последовательность чисел длиной m : $u_{m^*} = (u_{m^*}(x, 0), u_{m^*}(x, 1), \dots, u_{m^*}(x, m^* - 1)) \in U(x, m^*)$, зависящих от x .

Объект управления (ОУ) и система управления (СУ) взаимодействуют через коммуникационную сеть, в которой могут наблюдаться как случайные задержки в передаче данных, так и пропадание пакетов данных. Значения τ_{sc} , τ_c , τ_{ca} представляют собой задержку передачи данных между датчиками и системой управления (τ_{sc}); вычислительную задержку и задержку между СУ и частями системы (τ_c); задержку между ОУ и СУ (τ_{ca}), соответственно. Данные задержки могут моделироваться детерминистически, через определение верхних границ для наихудшего случая возможных задержек или стохастически, предполагая, что значения задержек подчиняются известному распределению.

Модельно-ориентированная компенсация сетевых эффектов. Идея метода модельно-ориентированной компенсации сетевых эффектов состоит в том, чтобы иметь математическое описание модели объекта управления на стороне системы управления и использовать данную модель для компенсации неопределенностей, вызванных неидеальной коммуникационной сетью, т.е. «внутренняя» модель объекта управления используется в системе прогностического управления для нейтрализации дестабилизирующих эффектов, вызванных задержками и пропажей пакетов.

Использование временных отметок и допущений о синхронизации важно для установления общего периода времени передачи пакета данных между СУ и ОУ. Хотя эти меры и носят ограничительный характер, они обычно используются в сетевых системах управления.

Подход, который предлагается использовать для повышения степени отказоустойчивости рассматриваемой системы управления, является детерминистическим. Это означает, что мы предполагаем знание верхних границ (для худшего случая) возможных задержек [1, 8].

Рассмотрим случай, когда присутствуют только задержки τ_{sc} . Поскольку нам заранее известна модель системы, и значение $x(k_s)$, то мы можем вычислить прогнозируемое значение $\tilde{x}(k_s + \tau_{sc})$ текущего состояния $x(k_s)$, основанного на старом измерении $x(k_s)$. Затем можно использовать прогнозируемое состояние для решения задачи оптимального управления в алгоритме СПУ.

Концепция согласованности предсказаний позволяет отделить анализ управления, например, с точки зрения устойчивости и качества, от влияния коммуникационной сети на исследуемый замкнутый контур. Предположим, что существует система управления, которая формирует допустимые последовательности $\mu_{m^*, \sigma} x(\cdot, \tilde{x}(\delta))$. Это может быть не только система, использующая СПУ, но и СУ любого другого типа.

Допустим, что система управления может стабилизировать систему независимо от того, сколько элементов каждой последовательности $\mu_{m^*, \sigma} x(\cdot, \tilde{x}(\delta))$ используются. Тогда её реализация через сеть с согласованными предсказаниями приведет к стабильной сетевой замкнутой системе. Причина этого заключается в том, что в отсутствие ошибок прогнозирования и согласованность прогнозов гарантирует совпадение прогнозируемого состояния с фактическим состоянием системы, поэтому последовательности обратной связи вычисляются на основе точной информации о состоянии системы.

Устойчивость модельно-ориентированной компенсации. В случае если одна из переданных управляющих последовательностей $\mu_{m^*, \sigma}$ не поступает или поступает слишком поздно в систему, значения управления в переданной ранее последовательности используются для обеспечения исполнительных механизмов текущей информацией [9, 10].

Поскольку временные задержки и потери пакетов варьируются в зависимости от текущей нагрузки сети, это означает, что длина $m \in \{1, 2, \dots, m^*\}$ каждой управляющей последовательности $\mu_{m^*, \sigma}$ может изменяться со временем. Это значение, теоретически, можно сделать независимым от времени, используя всегда значения $m = m^*$, однако из соображений надежности желательно всегда использовать самые последние управляющие входные данные, т.е. поддерживать m как можно меньшим.

Значит следует учитывать изменяющиеся во времени контрольные горизонты m , и поэтому необходимо найти условие, которое обеспечивает асимптотическую устойчивость замкнутого контура СПУ в этой ситуации. Для этого будем использовать функцию значения $V_N(\cdot)$, как общую функцию Ляпунова для неравенства:

$$V_N(x_{uN, m^*}(m, x) \leq V_N(x) - \alpha \sum_{k=0}^{m-1} \ell(x_{uN, m^*}(k, x), u_{N, m^*}(k, x)), \quad (7)$$

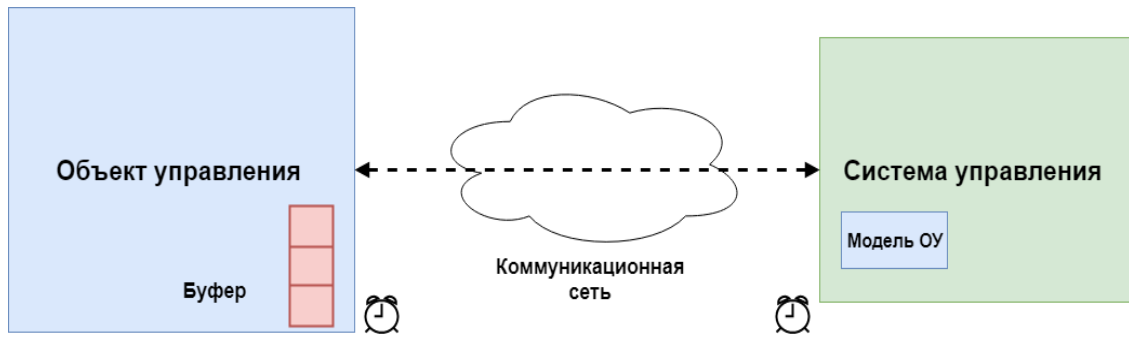


Рис. 3. Новая схема взаимодействия СУ и ОУ

где $\alpha \in (0,1]$ для каждого возможного состояния $x \in X$. Здесь $\mu_{m^*,\sigma}$ обозначает последовательность, состоящую из $m^* \in \{1,2,\dots,N\}$ вычисленных элементов $u^* \in U(x,N)$ на первом шаге алгоритма системы прогностического управления.

В дополнение к неравенству (7) зададим условия:

$$\eta_1 (\|x - x^*\|) \leq V_1(x),$$

$$V_N(x) \leq \eta_2 (\|x - x^*\|),$$

справедливые для всех $x \in X$. Тогда можно обеспечить асимптотическую устойчивость управления с помощью моделей прогнозирующего управления.

Для этого мы предполагаем, что для каждого возможного состояния $x \in X$ существует вектор управления $u \in U(x,\infty)$, а пара (C,ς) состоящая из $C \in \mathbb{R}$ и $\varsigma \in (0,1)$ удовлетворяет неравенству:

$$\ell(x_u(k,x),u(k)) \leq C\varsigma^k \quad \min_{u \in U(x,1)} \ell(x,u) = C\varsigma^k \ell^*(x).$$

Исходя из вышеизложенного, получим:

$$\alpha_{N,m} = 1 - \frac{\prod_{i=m+1}^N (\gamma_i - 1) \prod_{i=N-m+1}^N (\gamma_i - 1)}{\left(\prod_{i=m+1}^N \gamma_i - \prod_{i=m+1}^N (\gamma_i - 1) \right) \left(\prod_{i=N-m+1}^N \gamma_i - \prod_{i=N-m+1}^N (\gamma_i - 1) \right)}.$$

Внедрение предложенных решений в управление системой хранения данных «мягкое облачное хранилище»

Новая схема взаимодействия системы управления и объекта управления представлена на рис. 3. Теперь система управления посылает последовательность

управляющих значений, а не единичное значение, как это было раньше.

В объект управления был добавлен буфер, хранящий значения управления, которые будут применены в случае возникновения сетевых задержек и потери сетевых пакетов. Также система управления и объект управления были оснащены логикой работы с протоколом передачи данных, который обеспечивает передачу информации между компонентами системы вместе с временными метками.

Разработка инструмента для тестирования разработанных решений и его применение

Система управления СХД МОХ является программным обеспечением. Для тестирования такого рода ПО применяется метод «Внесения неисправностей», позволяющий улучшить тестирование программ, их отказоустойчивости и надежности путем внесения искусственных неисправностей разного рода.

Для разработки инструмента внесения этих неисправностей мы воспользовались двумя уже существующими компонентами – «Пумба» и «Куб-манки». Они являются открытым программным обеспечением и поэтому могут быть свободно использованы.

«Пумба» – популярный инструмент для генерации неисправностей, позволяющий завершать программные процессы в различных дистрибутивах *GNU/Linux* с различными кодами возврата. Также он может симулировать сетевые проблемы, такие как задержки и потери при передаче пакетов данных по сети.

«Куб-манки» позволяет удалять случайные контейнеры с запущенными приложениями, что помогает разработчикам проверять поведение их программ и создавать более надежное и отказоустойчивое программное обеспечение [1, 5, 10].

Наш инструмент будет запускать два компонента в автоматическом режиме, вносить искусственные неисправности и генерировать отчеты о результатах тестирования.

Алгоритм работы разработанного инструмента тестирования изображен на рис. 4.

Тестирование исходной системы управления. На рис. 5 изображен результат тестирования исходной системы управления. В момент времени 125.235 произошел сбой в коммуникационной сети. Исходная

СУ попыталась переконфигурировать систему, но потерпела неудачу, что привело к выходу системы из строя (временная отметка 171.536 и далее).

Тестирование новой модели системы управления. На рис. 6 изображен результат тестирования новой системы управления, использующей разработанную модель прогнозирующего управления. В момент времени 146.392 произошел аналогичный рассмот-

ренному выше сбой в работе коммуникационной сети. Однако новая модель системы управления, разработанная нами, смогла переконфигурировать систему благодаря СПУ и буферу, где хранились резервные управляющие последовательности.

После 189.392 система начала восстанавливаться и к временной отметке 209.513 вернулась к исходным характеристикам.

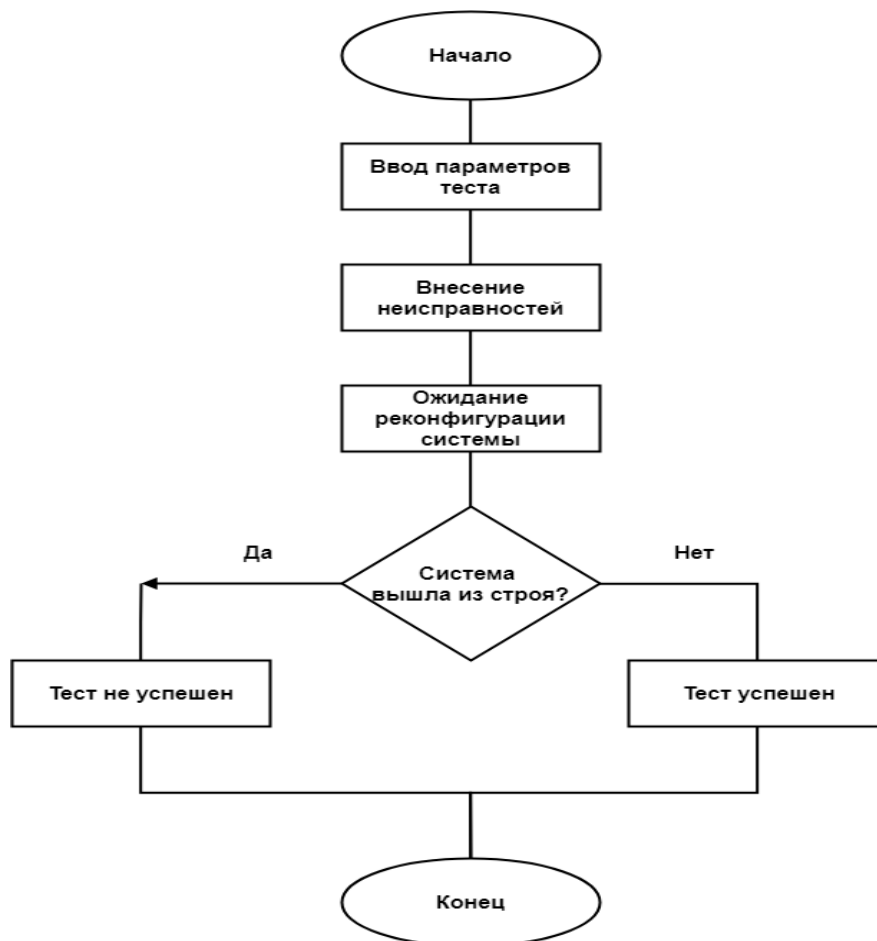


Рис. 4. Алгоритм работы инструмента тестирования

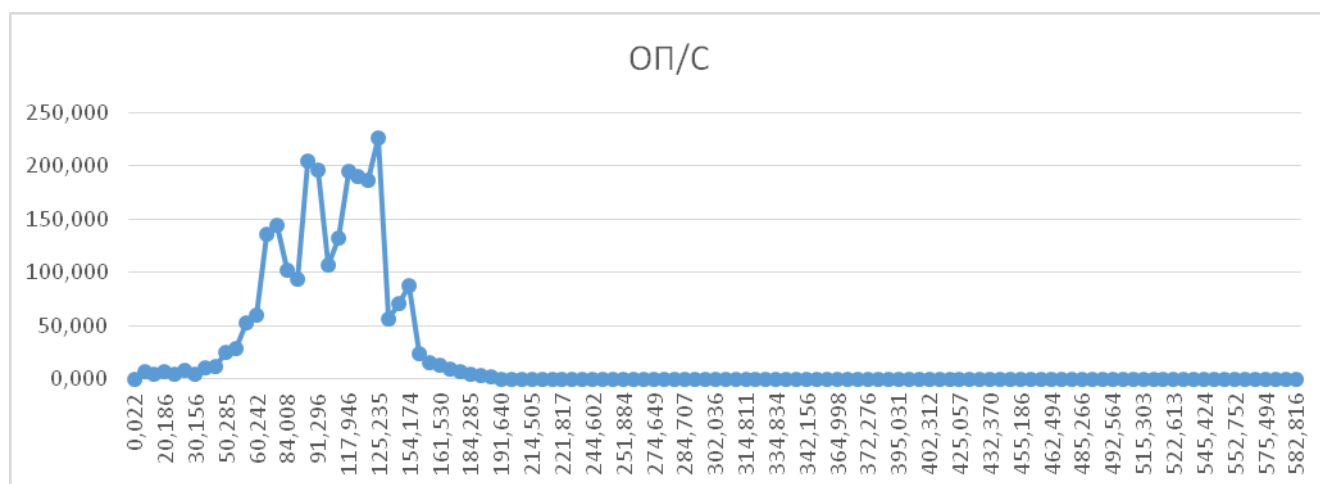


Рис. 5. Результаты работы исходной системы управления: по оси ординат отмечена текущая производительность СХД МОХ в операциях в секунду (ОП/с); по оси абсцисс – отсчеты времени

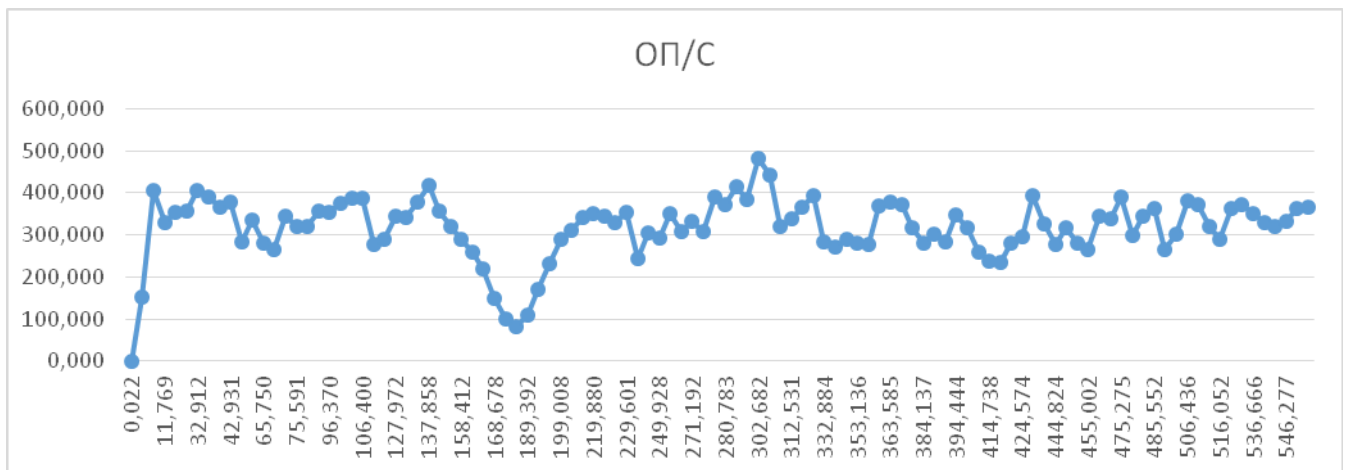


Рис. 6. Результаты работы новой модели системы управления: по оси ординат отмечена текущая производительность СХД МОХ в операциях в секунду (ОП/с); по оси абсцисс – отсчёты времени

ЗАКЛЮЧЕНИЕ

Рассмотрены методы повышения отказоустойчивости сетевых систем управления на примере системы хранения данных типа «мягкое облачное хранилище», представляющей собой программно-аппаратный комплекс. Для управления таким сложным объектом существует система управления, взаимодействующая с хранилищем через коммуникационную сеть. В таких сетях могут возникать временные задержки и потеря пакетов, что способно фатально повлиять на стабильность всей системы. В исходной версии системы управления не было механизмов борьбы с подобными сетевыми эффектами и их последствиями и, как следствие, система имела низкую отказоустойчивость.

Нами был предложен подход к обеспечению повышения степени отказоустойчивости системы хранения данных типа «мягкое облачное хранилище» в условиях неидеальной работы сети передачи данных, а также проанализированы результаты тестирования исходной системы управления и разработанной модели отказоустойчивого управления.

В соответствии с полученными результатами, можно сделать вывод, что предложенная нами модель может справляться с сетевыми задержками и потерей пакетов, в отличие от исходной. Выводы подтверждаются результатами экспериментов, которые проводились на функционирующей системе хранения данных «мягкое облачное хранилище».

СПИСОК ЛИТЕРАТУРЫ

- Zhilentov A.A., Nyrkov A.P., Chernyi S.G., Sokolov S. Simulation of In-Sensor Processes in the Sensor - Object System Type when Scanning the Elements of Underwater Communication Lines with a Probe Beam // International Review on Modelling and Simulations. – 2017. – Vol. 10, № 5. – P. 363-370
- Соколов С.С., Жилентков А.А., Черный С.Г. Обеспечение заданной степени устойчивости при управлении сложными системами с конечной точностью на примере мобильных объектов // Морская радиоэлектроника. – 2018. – № 4(66). – С. 42-45.
- Бланк М., Фрей, К. У., Краус К., Паттон Р.Дж., Старосветский М. Что такое отказоустойчивое управление // 4-й симпозиум IFAC по обнаружению неисправностей, надзору и безопасности для технических процессов: тезисы докл. Межд. конф. – Венгрия, 2000, С. 05579
- Алкасем А., Лю Х. Обзор отказоустойчивости в облачных вычислениях: концепции и практика // Исследовательский журнал прикладных наук, инженерии и технологии. – 2015. – №11. – С. 1365-1377.
- Дуброва Е. Отказоустойчивая архитектура. – 2-е изд. – NY: Springer, 2013. – 185 с.
- фон Нейман Дж. Вероятностная логика и синтез надежных организмов из ненадежного компонента // Исследования автоматов. – 1956. – №11. – С. 43-98.
- Цзян Ж., Чжао. К. Настраиваемое управление, основанное на неточной идентификации неисправностей // Труды Американской конференции по контролю. – USA, 1999. – С. 114-118.
- Гертлер Дж. Обнаружение и диагностика неисправностей в инженерных системах. - Нью-Йорк: Марсель Деккер, 1998. – 280 с.
- Джонсон Ф. Любопытство по поводу строк кода. // IT World. – URL: <https://www.itworld.com/article/2725085/curiosity-about-lines-of-code.html> (дата обращения: 27.04.2019)
- Nyrkov A.P., Zhilentov A.A., Sokolov S.S., Chernyi S.G. Hard- and Software Implementation of Emergency Prevention System for Maritime Transport // Automation and Remote Control. – 2018. – Vol. 79(1). – P. 195-202.

Материал поступил в редакцию 18.10.19.

Сведения об авторах

ЧЕРНЫЙ Сергей Григорьевич – кандидат технических наук, доцент, заведующий кафедрой электрооборудования судов и автоматизации производства Керченского государственного морского технологического университета, г. Керчь; доцент кафедры комплексного обеспечения информационной безопасности Государственного университета

морского и речного флота имени адмирала С.О. Макарова, Санкт-Петербург
e-mail: sergiiblack@gmail.com

ЖИЛЕНКОВ Антон Александрович – кандидат технических наук, доцент, заведующий кафедрой морской электроники Санкт-Петербургского государственного морского технического университета, Санкт-Петербург.
e-mail: zhilenkovanton@gmail.com

ВНИМАНИЮ ЧИТАТЕЛЕЙ!

ИЗДАНИЕ УДК

УНИВЕРСАЛЬНАЯ ДЕСЯТИЧНАЯ КЛАССИФИКАЦИЯ
АЛФАВИТНО-ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ
в 2-х томах

Алфавитно-предметный указатель (АПУ) к 4-му полному изданию УДК на русском языке:

Том I содержит АПУ от буквы А до Н;

Том II содержит АПУ от буквы М до Я и указатель латинских наименований к классам УДК 56 Палеонтология, 57 Биологические науки, 58 Ботаника, 49 Зоология, 61 Медицинские науки.

АПУ содержит около 100 000 понятий, представленных в полных таблицах УДК.

При его составлении были учтены изменения, опубликованные в Выпусках № 1 – 6 «Изменения и дополнения к УДК»

Для подписки необходимо направить заявку для оформления счета по адресу:

125190, Россия, Москва, ул. Усиевича, 20, ВИНТИ РАН

Телефоны: 499 155-42-85, 499 151-78-61

E-mail: feo@viniti.ru

<http://www.udcc.ru>