УДК 004.4

ОБЕСПЕЧЕНИЕ УСТОЙЧИВОСТИ И ЖИВУЧЕСТИ СПЕЦИАЛИЗИРОВАННЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В КОРПОРАТИВНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

Н. В. Стецюк[1], В. Н. Стецюк[2], О. С. Савенко[3]

[1] *Аспирант кафедры компьютерной инженерии и системного программирования Хмельницкого национального университета Хмельницкий, Украина*

[2] *Д. т. н., профессор, декан факультета программирования, компьютерных и телекомуникационных систем Хмельницкого национального университета, Хмельницкий, Украина*

[3] *Начальник отдела информационно-технического обеспечения экономических служб, старший преподаватель кафедры компьютерной инженерии и системного программирования Хмельницкого национального университета, Хмельницкий, Украина*

Реферат

Разработан подход к определению эффективности ИТ на основе учета количественных величин, характеризующих отказоустойчивость и живучесть, который может быть расширен для учета других характеристических величин. Для обеспечения отказоустойчивости и живучести ИТ разработана система мероприятий в результате выполнения которых получено ИТ узкоспециализированного использования для различных сфер применения, где сопровождаемые процессы относятся к системам ирреального или нереального времени с достаточно высокими параметрами отказоустойчивости, живучести и вообще резилентности и, в то же время, приемлемыми финансовыми затратами на ее эксплуатацию.

Ключевые слова: отказоустойчивость ИТ, живучесть ИТ, резилентность, компьютерные сети.

ENSURING THE STABILITY AND SURVIVABILITY OF SPECIALIZED INFORMATION TECHNOLOGIES IN CORPORATE COMPUTER NETWORKS

N. V. Stetsyuk, V. N. Stetsyuk, O. S. Savenko

Abstract

An approach to determining the effectiveness of information technology based on quantitative values that characterize its fault tolerance and survivability under the influence of malicious software, and can be extended to take into account other characteristics. To ensure fault tolerance and survivability of IT, a system of measures has been developed, which resulted in highly specialized IT applications for various applications, where the accompanying processes are unrealistic or unrealistic time with high parameters of fault tolerance, survivability and overall resilience and, at the same time, acceptable equal financial costs for its operation.

Keywords: Software, Fault tolerance, Resilience, Survivability, Computer.

Введение

Для информационных технологий (ИТ), которые обеспечивают жизнедеятельность учреждений или предприятий в различных специализированных областях, вопросы живучести и отказоустойчивости являются очень важными, особенно вследствие роста их количественных параметров функционирования (увеличение пользователей, серверов, объемов информации в базах данных) и уровня сложности.

От обеспечения этих параметров в прямой зависимости находится эффективность функционирования всей ИТ. Для специализированных ИТ, функционирующих в корпоративных компьютерных сетях и выполняющих функцию информационного обеспечения в такой узкоспециализированной предметной области, как финансово-хозяйственная деятельность в различных сферах применения, этот параметр значительно выше нуля, но требования к таким ИТ тоже достаточно высоки, которые невозможно выполнить при низких параметрах отказоустойчивости и живучести, особенно при постоянном росте числа пользователей, увеличении сложности информационных потоков и объемов обрабатываемых данных.

Обеспечение высокой эффективности специализированных ИТ осуществляется на основе реализации в них принципов отказоустойчивости и живучести, что является актуальной научной задачей, которая начинает решаться еще в процессе разработки специализированных ИТ.

Анализ известных работ

Известные методы обеспечения отказоустойчивости и живучести специализированных ИТ ориентированы на их различные типы, приложения, функции компьютеров и особенности реализации в различных компонентах ИТ. Кроме того, неотложной областью, требующей исследования, является влияние внешних факторов (распределенные атаки, вредоносное программное обеспечение (ПО)) на функционирование ИТ, устойчивость и живучесть.

В [1] предложена информационная технология для оценки структурной надежности технических объектов, структура которых соответствует одному из известных типов нейронных сетей. Структура информационных технологий содержит морфологическую модель, которая позволяет формировать и изменять структуру модели исследуемого объекта по правилам, основанным на определении вероятности безотказной работы в теории реальности. Разработанная модель позволяет модифицировать объектную модель с учетом отказоустойчивости ИТ.

В [2–4] рассматривается надежность информационных систем. Подход, основанный на оценке рисков и снижении рисков, используется для повышения надежности информационных систем. Такой подход позволяет на ранней стадии оценить риск процесса разработки программного обеспечения и определяет наиболее эффективные стратегии снижения риска.

В статьях [5, 6] исследуются облачные приложения, которые относятся к компонентам нескольких облачных сервисов, которые взаимодействуют друг с другом через интерфейсы веб-сервисов, где каждый компонент выполняет определенные функциональные возможности. Отсутствие эффективной схемы отказоустойчивости является одним из основных препятствий на пути повышения доступности и эффективности сложных облачных систем для развертывания.

doi.org/10.36773/1818-1212-2020-123-5-58-65

В [7–9] представлены подходы для предотвращения функциональных сбоев во время выполнения в компонентных прикладных система, использующих внутреннюю избыточность компонентов для поиска обходных путей в виде альтернативных последовательностей операций во избежание сбоев. Предложена проактивная схема восстановления на основе миграции сервисов, описаны толерантные системы. Рассмотрены методы обеспечения отказоустойчивости. Обсуждаются существующие методы обеспечения устойчивости облачных вычислений на основе их политик, используемых инструментов и проблем исследования.

В [10] рассматривается киберживучесть систем. Наиболее распространенные методы обеспечения отказоустойчивости с использованием методов избыточности.

Известные методы и методы обеспечения отказоустойчивости и живучести специализированных ИТ недостаточно систематизированы и не всегда могут быть реализованы из-за специфики использования и структуры специализированных ИТ. Следовательно, необходимы дальнейшие исследования и разработка новых методов и технологий, которые могут повысить отказоустойчивость и живучесть специализированных ИТ, включая кибератаки и вредоносное ПО.

Критерии эффективности и устойчивости специализированных информационных технологий в корпоративных компьютерных сетях

Представим специализированную ИТ в корпоративных компьютерных сетях множеством ее компонентов:

$$S_{IT} = \{S_1, S_2, ..., S_n\}, \quad (1)$$

где S_i – компонента специализированной ИТ в корпоративных компьютерных сетях, $i = 1,2,...,n$, n – количество компонентов.

Для каждой компоненты S_i применим функцию, которая будет включать все критерии эффективности в корпоративных компьютерных сетях, применение которых при разработке ИТ необходимо для дальнейшего пользования ею. В частности, среди таких критериев будут также критерии обеспечения отказоустойчивости и живучести. Зададим критерии эффективности специализированных ИТ вектором, компонентами которого будут функции эффективности, соответствующие конкретным критериям:

$$K_e = (f_1, f_2, ..., f_m), \quad (2)$$

где f_j – функция, которая задает один из критериев эффективности, $j = 1,2,...,\mathrm{m}$, m – количество функций.

Учитывая то, что в целом задача достижения максимальной эффективности зависит от конкретных критериев, которые могут быть связаны между собой и, соответственно, влиять друг на друга, при этом улучшение эффективности одного может привести ухудшению другого. Кроме того, поскольку специализированные ИТ состоят из компонентов, к которым применяются те же критерии из заданного вектора, то задача усложняется тем, что часть компонентов ИТ различна и, соответственно, достижения эффективности по тем же наборами критериев будет различной. Поэтому выбор оптимальных решений является сложной многокритериальной задачей. Общую постановку задачи поиска лучшей эффективности для специализированных ИТ в корпоративных компьютерных сетях сформулируем так:

$$\begin{cases} K_e(S_{IT}) \to max; \\ f_j(S_i) \to max,\ i = 1,2,...,n;\ j = 1,2,...,m. \end{cases} \quad (3)$$

Кроме того, некоторые ИТ-компоненты могут быть функционально повторяющимися в зависимости от задач и места нахождения в корпоративных компьютерных сетях. Это повлияет на общую эффективность специализированных ИТ. Однако достижение производительности по определенным критериям в одних и тех же компонентах специализированной ИТ не обязательно должно быть одинаковым, потому что эти компоненты будут решать разные задачи или одни и те же задачи, но в разное время они будут проходить разные стадии. Помнить об этих функциях важно, поэтому мы детализируем задачу по поиску наилучшей производительности для специализированных ИТ в корпоративных компьютерных сетях следующим образом:

$$\begin{cases} K_e(S_{IT}) \to max; \\ f_{j,q}(S_{i,p}) \to max,\ i = 1,2,...,n;\ j = 1,2,...,m \\ q = 0,1,...,n_q;\ j = 0,1,...,n_p \end{cases} \quad (4)$$

где q – номер компонента специализированной ИТ в определенном узле корпоративной компьютерной сети; j – индекс для критерия эффективности компонентов специализированных ИТ в определенном узле корпоративной компьютерной сети; $q = 0,1,...,n_q$, $j = 0,1,...,n_p$; n_q – количество одинаковых компонентов специализированной ИТ в корпоративной компьютерной сети; n_p – номер критерия для одинаковых компонентов специализированной ИТ в корпоративной компьютерной сети.

Введем функцию, которая будет определять максимальное значение критерия эффективности:

$$F : K_e(S_{IT}) \to \max. \quad (5)$$

Значение критерия эффективности зададим выражением с учетом весовых коэффициентов:

$$K_e(S_{IT}) = \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{p=0}^{n_p} \sum_{q=0}^{n_q} \left(\alpha_{i,j,p,q} \cdot f_{j,q}(S_{i,p}) \right), \quad (6)$$

где $\alpha_{i,j,p,q}$ – весовые коэффициенты.

Рассмотрим достижения максимизации критериев по показателям отказоустойчивости и живучести в конфигурациях информационных технологий, построенных на основе архитектуры «клиент – сервер» с их обеспечением по всем звеньям системы от пользователей (клиентская часть) к критически важной серверной части. Выбор для рассмотрения именно архитектуры «клиент – сервер» зависит от ее особенностей, которые проявляются в следующем: базовые функции клиентского приложения распределяются между клиентом и сервером; программное обеспечение автоматизированного рабочего места клиентского компьютера работает с данными через запросы к серверному программному обеспечению; осуществляется полная поддержка многопользовательской работы; гарантируется целостность данных. Это отличает ее от других архитектур и позволяет осуществить обеспечения отказоустойчивости и живучести в каждой из звеньев системы отдельно.

Основными направлениями для повышения живучести и отказоустойчивости ИТ является внесение избыточности в конфигурацию аппаратных и программных средств, поддерживающей инфраструктуры, резервирование информационных ресурсов (программ и данных). Известны два подхода в построении отказоустойчивой ИТ. Первый подход базируется на использовании отказоустойчивых компонентов. Такая ИТ обеспечивает свои функции даже при выходе из строя подкомпонентов некоторых компонентов. Это самый простой метод, но вместе с тем и самый дорогой, через применение самых доро-

гих составляющих – отказоустойчивых компонентов ИТ. Второй способ заключается в построении отказоустойчивой ИТ с использованием компонентов, которые не являются отказостойкими. Отказоустойчивость в таких системах достигается за счет введения в них избыточности через резервирование критических звеньев аппаратного обеспечения, программного обеспечения, межкомпонентных связей и специальных алгоритмов функционирования ИТ, предусматривающие ее реконфигурацию при отказе некоторых компонентов.

Главным свойством отказоустойчивости является прозрачность отказов ее отдельных компонентов для конечного пользователя. Это означает, что отказоустойчивая система автоматически меняет свою конфигурацию в случае отказа. Ее программное обеспечение в процессе выполнения ищет обходные пути, пытаясь в условиях отказа привести выполняемую функцию до успешного завершения.

Зададим функцию $f_1(S_i)$, $i = 1,2,\ldots,n$ определение отказоустойчивости в компьютерных системах в количественном виде так:

$$f_1(S_i) = \frac{T_{f_1(S_i),1}}{T_{f_1(S_i),1} - \left(T_{f_1(S_i),2} - T_{f_1(S_i),3}\right)}, \qquad (7)$$

где i – количество компонентов специализированной ИТ, $i = 1,2,\ldots,n$, $T_{f_1(S_i),1}$ – время между соседними сбоями; $T_{f_1(S_i),2}$ – время, необходимое для обнаружения сбоя и поиска пути его обхода; $T_{f_1(S_i),3}$ – время, необходимое для восстановления ИТ после сбоя.

Как видно из формулы (7), для ИТ с автоматической системой обеспечения отказоустойчивости она будет приближаться к максимуму из-за скорости реакции. Для построения таких систем нет теоретических препятствий, но на практике при их реализации нужно учитывать ряд важных факторов: финансовые издержки реализации автоматической системы обеспечения живучести и отказоустойчивости; сложность системы. Для ИТ, предназначенных для информационного обеспечения в узкой специализированной предметной области, например финансово-хозяйственная деятельность учреждения высшего образования, будет целесообразным отказаться от автоматической системы управления отказоустойчивостью в пользу автоматизированной. При таком подходе часть дорогих функций управления резервированием, присутствующими в ИТ, будет возложена на человека. Тогда, согласно формуле (7), отказоустойчивость $f_1(S_i)$ будет ниже, чем в первом случае. Но решением задачи построения ИТ (подобно тому, как и в других задачах проектирования) является не обеспечение максимально возможной отказоустойчивости системы, а нахождение приемлемого баланса параметров системы в рамках определенного технологического базиса. Исследуем решения вопросов обеспечения отказоустойчивости ИТ при использовании такой стратегии. Проанализируем факторы, негативно влияющие на отказоустойчивость ИТ со стороны клиента. Схема воздействия негативных факторов на видмостийкисть клиентской части специализированной ИТ изображена на рис. 1.

Как видно из предложенной модели, негативные факторы, влияющие на отказоустойчивость клиентской части ИТ, делятся на внешние и внутренние. Среди внешних факторов наибольшую угрозу представляют собой сбои в работе энергосистем питания и природные явления, которые могут привести к отказам компонентов компьютеров и компьютерных сетей.

Другим важным фактором является ошибки в коде системного программного обеспечения. Причина в том, что системное программное обеспечение представляет собой сложную систему и каждое исправление ранее найденной ошибки не гарантирует отсутствия привнесения новой. Этот аспект, связанный с системным программным обеспечением, является порождением еще одного фактора, который может уменьшить отказоустойчивость. В силу его сложности при настройке программного обеспечения могут вноситься ошибки настройки. Уменьшить его проявление можно путем использования программного обеспечения с автоматической настройкой, что не всегда приемлемо, и привлечением более квалифицированного персонала.

Для прикладного программного обеспечения, к которому относятся клиентские части специализированной ИТ, критические ошибки, которые проявились в ходе эксплуатации

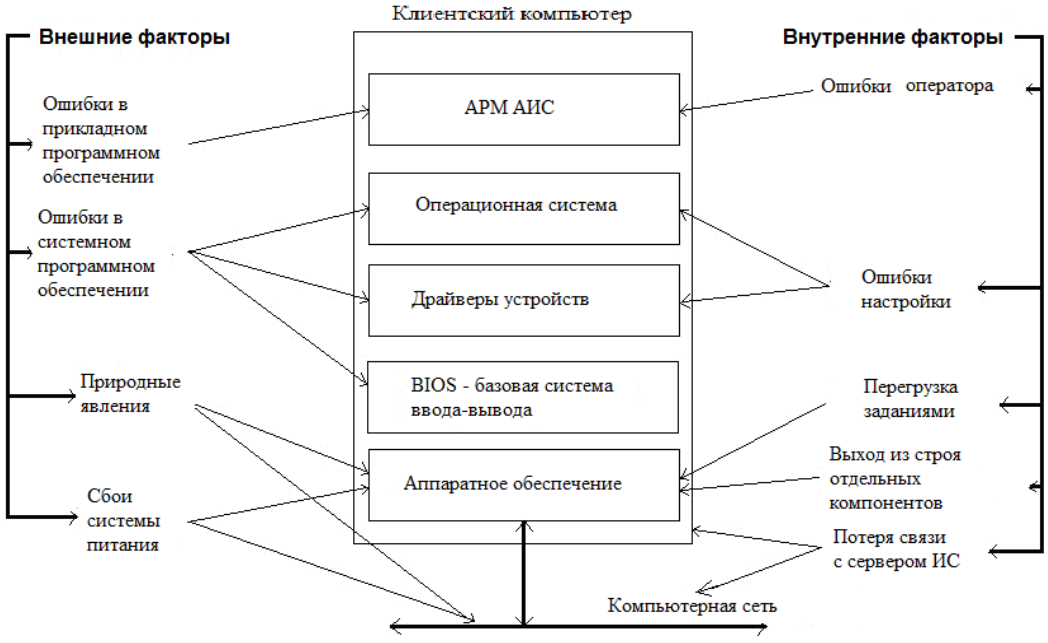
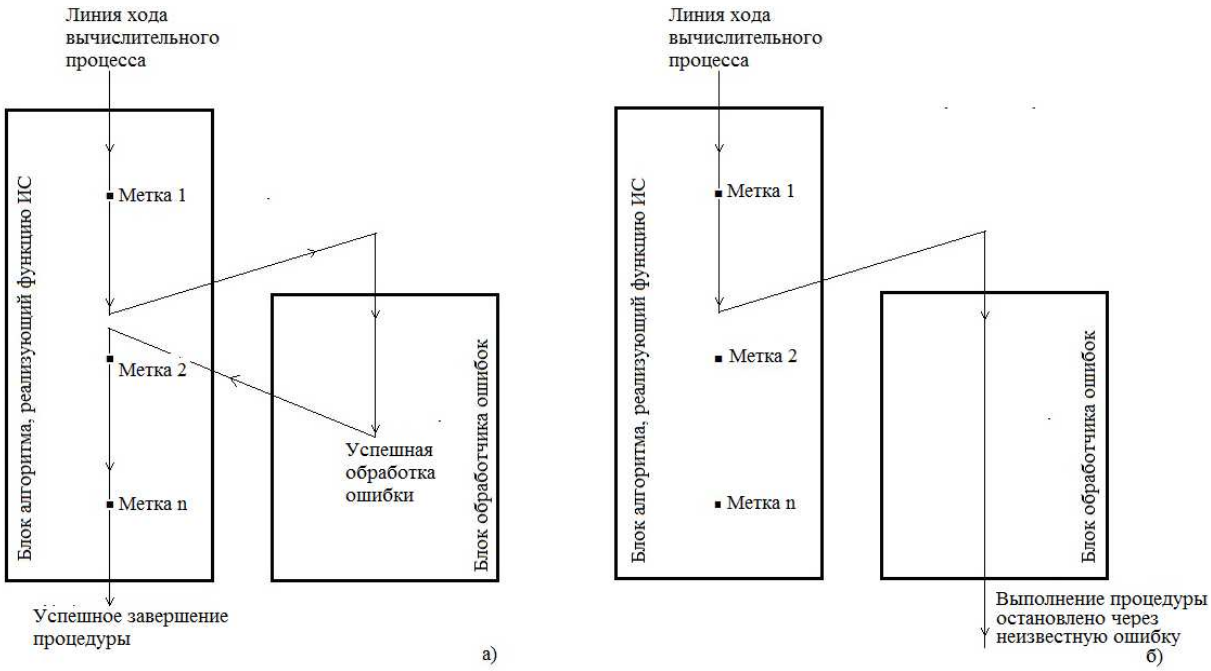


Рисунок 1 – Схема действия негативных факторов, влияющих на отказоустойчивость клиентской части специализирован-

doi.org/10.36773/1818-1212-2020-123-5-58-65



а) для случая успешного завершения процедуры; б) для случая, когда ошибка неизвестна для обработчика ошибок

Рисунок 2 – Модель работы алгоритма отказоустойчивой процедуры

рабочих мест, фиксируются вместе со своими параметрами в реестре системы в автоматический способ и в дальнейшем используются для анализа с целью устранения причин, которые вызвали. Это достигается благодаря подходу, основанному на привнесении некоторой избыточности в программное обеспечение клиентской части ИТ. С этой целью все расчетные процедуры, которые могут содержать критические для функционирования ошибки, разработаны с соблюдением заданого однотипного шаблона построения алгоритма ее выполнения. Суть алгоритма отражена на рис. 2.

В данной структуре алгоритм выполнения любой нетривиальной процедуры разделяется на два взаимодействующих блока. В первом блоке реализуется функция процедуры ИТ, а во втором – обработчик ошибок. В процессе выполнения некоторой процедуры, которая реализует одну из функций компонентов ИТ, оба блока взаимодействуют между собой, передавая управление вычислительным процессом друг другу, пока выполняемая функция не завершится. Его суть заключается в том, что алгоритм, который реализует функцию ИТ, разделяется маркерами (метка 1, ..., метка *n* на рис. 2) на фрагменты по принципу функциональной завершенности.

Такое решение, кроме всего, позволяет собирать информацию о фатальных ошибках, которые произошли в процессе функционирования ИТ, и в процессе дальнейшего анализа позволяет выявить слабые звенья в ИТ с целью их устранения путем усовершенствования программного обеспечения клиентской части ИТ.

Программное обеспечение клиентских частей ИТ на протяжении жизненного цикла ИТ по разным причинам, в том числе и через обнаружении в нем ошибки, может изменяться, проходя свои собственные циклы обновления.

Следующим из значимых внутренних факторов, негативно влияющих на отказоустойчивость, является перегрузка аппаратной платформы клиентского компьютера задачами, которая может резко ухудшить временные параметры выполняемых задач клиентской частью ИТ, или даже сделать невозможной его работу из-за исчерпания технических ресурсов. Чтобы нейтрализовать действие этого фактора в ИТ, при разработке программного обеспечения применено функциональное резервирование, а именно той его части, которая ответственна за реализацию "бизнес-логики".

Наличие функционального резерва "тяжелых" расчетных функций позволяет осуществлять маневр вычислительными мощностями аппаратной платформы ИТ в случае перегрузки отдельных ее звеньев, повышая таким образом отказоустойчивость. Поскольку процедура, которая функционально резервируется, разрабатывается в двух вариантах по одному алгоритму, но в разных программных средах, для выполнения в различных технических средствах. В этом проявляется положительная мультипликативность эффекта функционального резервирования, что повышает общую отказоустойчивость ИТ.

В целом, отказоустойчивость клиентской части обеспечена путем выполнения комплекса мероприятий, включающих как перечисленные выше, так некоторые дополнительные. Например, использование нетривиальных редакторов данных, включающих в свой алгоритм работы интерактивную процедуру, что исключает неконтролируемое манипулирование данными базы данных (БД) со стороны оператора.

В результате анализа факторов, базы данных которых негативно влияют на отказоустойчивость серверной части ИТ, была построена модель действия негативных факторов, изображенная на рис. 3.

В представленной модели все негативные факторы делятся на внешние, вызванные причинами, которые находятся за пределами системы, и внутренние. Внешние факторы, которые уменьшают отказоустойчивость серверной части, нейтрализуются тем же способом, что и в клиентской части ИТ. Но в силу своей значимости этого недостаточно. Так как сервер ИТ является местом нахождения базы, где сосредоточена вся информация, обрабатываемая в системе, то для него такой фактор, как сбои в системе питания, особенно опасен. Это связано с тем, что БД, будучи сама по себе сложно организованной системой, является чувствительной к нарушению технологии обращения с ней. Внезапное пропадание питания или выход из

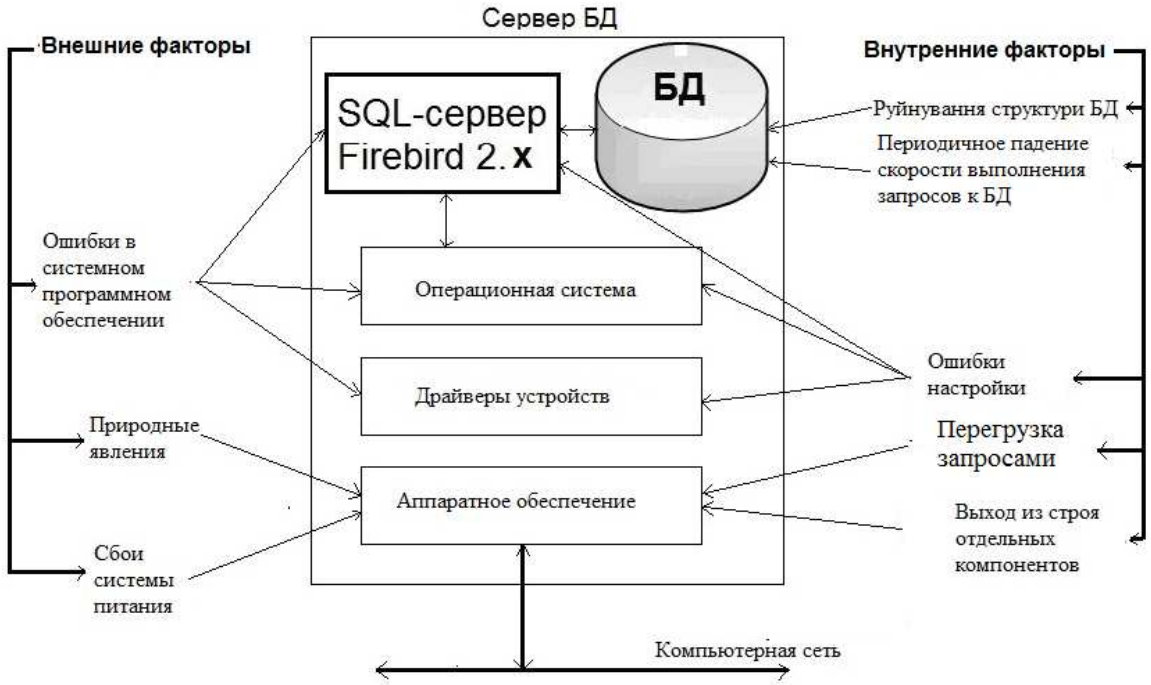


Рисунок 3 – Модель действия негативных факторов, влияющих на отказостойкость сервера ИС

строя аппаратного обеспечения сервера в силу флуктуаций напряжения может привести с высокой вероятностью к повреждению базы данных со всеми последующими негативными последствиями для информации. С целью недопущения такого развития событий, в контур системы питания было введено устройство бесперебойного питания с двойным преобразованием напряжения и достаточным временем обеспечения автономной работы сервера. Кроме того, устройство бесперебойного питания должен иметь контроллер состояния, который включает в себя выход с последовательным интерфейсом. Он необходим для передачи серверу сигнала разгрузки в случае, когда из-за длительного пропадания внешнего питания, будет исчерпан до недопустимого предела внутренний запас электроэнергии устройства бесперебойного питания. По этому сигналу сервер корректно завершит все приложения, которые были запущены, не допустив разрушения БД.

Не менее угрожающими для серверной части, которые уменьшают ее отказоустойчивость, есть внутренние факторы. Среди них наиболее тяжелый по последствиям выход из строя аппаратных средств, а именно накопителей. Для сервера ИТ они являются самым ответственным звеном, отказ которого может привести не только к длительной недоступности к информационным ресурсам, но и безвозвратной потере данных. Поскольку потеря БД является неприемлемой, то необходимы меры для нейтрализации угрозы внезапной потери накопителя. Эта задача может быть решена путем его резервирования. Вместо отдельного накопителя для хранения БД и других критических данных может быть применен RAID массив накопителей типа 1. С точки зрения оценки дополнительных расходов на резервирование стоимость дополнительного накопителя небольшая в сравнении с возможными потерями, вызванными потерей БД. Кроме того, организация периодического диагностирования накопителей позволит в большинстве случаев заблаговременно выявлять проблемы накопителя.

Еще одним способом недопущения потери БД является организация ее резервного копирования. Несмотря на описанные меры обеспечения отказоустойчивости серверной части ИТ, они все же не могут претендовать на абсолютность. Поскольку БД и вся последовательность программного обеспечения, обеспечивающего ее работу, является сложной системой, то наличие ошибок ее функционирования остается достаточно высоким. Чтобы уменьшить воздействие таких деструкций, как неизвестные ошибки, которые могут привести к разрушению БД, в контур программного обеспечения сервера можно включить подсистему резервного копирования БД. Она работает в автоматическом режиме согласно графику. Репозитарием копий БД служит другой компьютер, территориально удаленный от основного сервера. Поскольку копия БД является достаточно большим массивом информации, поэтому, чтобы не зависеть от сетевого трафика, основной сервер и компьютер с репозитарием копий БД имеют собственный канал связи. Для недопущения неконтролируемых изменений данных в БД, которые могут нарушить целостность данных ИТ, все изменения выполняются под управлением транзакций. Такой подход обеспечивает переход БД из одного согласованного состояния в другое при манипулировании данными.

Важно подчеркнуть, что процесс обеспечения отказоустойчивости является непрерывным на протяжении всего жизненного цикла ИТ. Он начинается с планирования мер обеспечения отказоустойчивости ИТ, проектируется и продолжается до момента завершения ее функционирования вообще.

Показатели живучести в сложной системе: многофункциональность отдельных компонентов; наличие единой (главной) цели функционирования всей системы; возможность не только информационного обмена между отдельными компонентами, но и информационного взаимодействия с пользователями; наличие средств защиты, контроля, диагностики и самоорганизации. Задача анализа структурной живучести требует определения: системной архитектуры, необходимой для выполнения цели функционирования ИТ в некоторый момент или промежуток времени, когда возникают нежелательные воздействия на систему; требований по отдельным видам ресурсов системы и их взаимосвязи; требований по функцио-

doi.org/10.36773/1818-1212-2020-123-5-58-65

нальным возможностям ресурсов системы; особенностей характера нежелательных воздействий или их последствий.

Зададим функцию $f_2(S_i)$, в которой $i = 1,2,...,n$ определения живучести в количественных единицах компьютерных сетях выразим так:

$$f_2(S_i) = \frac{T_{f_2(S_i),1} - T_{f_2(S_i),2}}{T_{f_2(S_i),1}}, \quad (8)$$

где $T_{f_2(S_i),1}$ – время функционирования процессе ИТ в стандартном режиме работы; $T_{f_2(S_i),2}$ – время, затраченное на процессы обеспечения живучести, $i = 1,2,...,n$.

Такое определение функции живучести позволяет отобразить стандартный режим работы значением единицы, а при возникновении потребности в обеспечении живучести и в случае намного более длительного времени, чем время стандартного режима работы, значение функции отражать количественную порядковую величину.

Резервирование сервера гарантирует достаточную живучесть ИТ в целом, но не дает гарантии потери ею некоторых своих функций, связанных с выходом из строя аппаратных компонентов клиентского компьютера, критически влияющих на функционирование клиентской части в целом. Решение задачи, в таких случаях, заключается в создании определенного резерва. Его особенностью является то, что в качестве резерва здесь служит любой другой клиентский компьютер, который согласно плану преодоления критической ситуации, может взять на себя обеспечение работы программного обеспечения клиентской части, чей компьютер вышел из строя.

При этом модули программного обеспечения в настроенном виде хранятся в репозитарии программного обеспечения ИТ и на тех клиентских компьютерах, где они планируются быть использованы в критические моменты согласно плану резервирования. В случае выхода из строя критического оборудования компьютера, оно переносится на другой компьютер, согласно плану резервирования.

Такая реконфигурация клиентской части стало возможной благодаря тому, что на клиентских компьютерах, на которых выполняется программное обеспечение, не сохраняются абсолютно никакие данные. А сам программный модуль, для удобства скомпонованный в один файл, и не требует процедуры инсталляции. Ее достаточно скопировать на другой компьютер, после чего она будет готова к работе. Есть только одно ограничение – каждый экземпляр программного обеспечения клиентской части предварительно должен быть зарегистрирован в ИТ. Иначе попытка запуска такой программы будет рассматриваться как попытка несанкционированного доступа к системе, даже при правильных регистрационных данных. Контроль ИТ по всем экземплярам своих клиентских частей позволяет блокировать попытки злоумышленников, которым удалось овладеть данным аккаунтаом пользователя, получить доступ к системе. При этом программа, которой овладел злоумышленник, не получает доступа к данным ИТ, а сам факт попытки такой программы подключиться к системе фиксируется в реестре фатальных ошибок с данными, что позволяет их использовать для принятия организационные мер против злоумышленников.

Таким образом, живучесть ИТ обеспечивается резервированием серверной части ИТ с территориальным разнесением основного и резервного сервера, резервированием программного обеспечения клиентской части. Особенностью резервирования является то, что в качестве резерва служат не аппаратные модули, а резерв производительности отдельных клиентских компьютеров, которые, согласно плану резервирования в критический момент будут использоваться как штатные, не допуская потери функциональности ИТ.

На основе формул (6)–(8) получим значение эффективности для ИТ с учетом показателей отказоустойчивости и живучести:

$$K_e(S_{IT}) = \sum_{j=1}^{m}\sum_{p=0}^{n_p}\sum_{q=0}^{n_q}\left(\alpha_{1,j,p,q} \times \right.$$
$$\left. \times \frac{T_{f_1(S_i),1}}{T_{f_1(S_i),1} - \left(T_{f_1(S_i),2} - T_{f_1(S_i),3}\right)} + \alpha_{2,j,p,q} \cdot \frac{T_{f_2(S_i),1} - T_{f_2(S_i),2}}{T_{f_2(S_i),1}}\right), \quad (9)$$

где $\alpha_{1,j,p,q}$ – коэффициент для значения, которое определяет отказоустойчивость в количественных единицах; $\alpha_{2,j,p,q}$ – коэффициент для значения, которое определяет живучесть в количественных единицах; $\alpha_{1,j,p,q} + \alpha_{2,j,p,q} = 1$.

Аналогично, слагаемыми в формуле (6) и ее конкретизации формуле (9) для двух величин могут быть другие показатели, характеризующие эффективность ИТ.

В результате использования перечисленных мероприятий было получено ИТ узкоспециализированного использования для различных сфер применения, где процессы сопровождения относятся к ирреальному или нереальному времени с достаточно высокими параметрами отказоустойчивости, живучести и вообще резилентности и в то же время приемлемым относительно финансовых затрат на ее эксплуатацию.

Эксперименты и оценка

Для определения, насколько эффективны предлагаемые решения по обеспечению отказоустойчивости и живучести, проведем сравнение критерия эффективности для ИТ без обеспечения отказоустойчивости и живучести и с включением этих характеристик на основе формулы (9).

Значение величины критерия эффективности ИТ, в которой не обеспечиваются требования отказоустойчивости и живучести, получаем из формулы (9) так: 1) решение проблем, связанных с отсутствием обеспечения в ИТ реализованных отказоустойчивости и живучести, возложена на оператора или администратора, который постоянно мониторит функционирования ИТ; решение проблемных ситуаций осуществляется только при их обнаружении. В первом случае расчет по формуле (9) может быть аналогичным и получим значения величины на порядок выше значения критерия для ИТ, где обеспечивается отказоустойчивость и живучесть. Если же рассматривать второй вариант, тогда $K_e(S_{IT}) = 1$. В этом случае отношение между значениями определяется по формуле (10) и позволяет установить эффективность предложенных решений по обеспечению отказоустойчивости и живучести, а также улучшить достижения эффективности за счет корректировки коэффициентов:

$$\mu = \frac{1}{\sum_{j=1}^{m}\sum_{p=0}^{n_p}\sum_{q=0}^{n_q}\left(\alpha_{1,j,p,q} \cdot \frac{T_{f_1(S_i),1}}{T_{f_1(S_i),1} - \left(T_{f_1(S_i),2} - T_{f_1(S_i),3}\right)} + \alpha_{2,j,p,q} \cdot \frac{T_{f_2(S_i),1} - T_{f_2(S_i),2}}{T_{f_2(S_i),1}}\right)}, \quad (10)$$

где отсутствие сбоев в работе специализированной ИТ или внешних воздействий означает, что время, потраченное на их обработку равно нулю и соответственно отношение станет равным единице.

doi.org/10.36773/1818-1212-2020-123-5-58-65

Log_reconfig.log

```
16  Apr 15 09:01:02 itz0 run-parts(/etc/cron.hourly)[17261]: starting 0anacron
17  Apr 15 09:01:02 itz0 run-parts(/etc/cron.hourly)[17270]: finished 0anacron
18  Apr 15 10:13:16 itz0 CROND[17274]: (root) CMD (/Stecjk/db-hourly)
19  Apr 15 11:43:16 itz0 sshd[30314]: False error in the operation of the
20        network device from 192.168.168.2 port 43760 ssh2
21  Apr 15 11:44:01,786 DEBUG :NetworkDevice eth0:
22        DEVICE="eth0"
                              . . .
32        TYPE="Ethernet"
33        IPV4_FALSE_MISTAKE=yes
34        UUID="ee9c32a3-47c2-4217-b817-82e1d91f6a5f"
35  Apr 15 11:44:12 itz0 sshd[29043]: pam_unix(sshd:session): session closed
36        for user swm
37  Apr 15 11:44:56 itz0 sshd[29078]: Network device configuration required ...
38  Apr 15 11:46:02,786 DEBUG   : writeIfcfgFile eth1
39        to /etc/sysconfig/network-scripts/ifcfg-eth0 not needed
40  Apr 15 11:46:21,396 DEBUG   : Network.write() called
41  Apr 15 11:46:21,397 DEBUG   : /etc/sysconfig/network-scripts/ifcfg-eth1:
42        DEVICE=eth1
43        TYPE=Ethernet
                              . . .
50        PREFIX=24
51        DEFROUTE=yes
52  Apr 15 11:47:41 itz0 sshd[30314]: pam_unix(sshd:session): session opened for user swm by (uid=0)
53        IPV6INIT=no
```

Рисунок 4 – Фрагмент Log-файла подсистемы контроля работы сетевых устройств

Если же произойдет сбой или внешнее вмешательство, тогда значение будет больше единицы. Эффективным значением является значение, минимально отклоненное от единицы.

Результаты обеспечения отказоустойчивости и живучести специализированной ИТ изображены в реализованной ИТ на рис. 4.

Для удобства все строки фрагмента log-файла были пронумерованы, а критические позиции выделены.

В позиции 19 обнаружена роковая ошибка в работе сетевого адаптера «eth0» в момент обращения пользовательского компьютера с IP 192.168.168.2. В позиции 35,36 закрывается текущая сессия пользователя SWM. В позиции 37 система сообщает, что нужна реконфигурация сетевых устройств. В позиции 38 сообщается, что устройство «eth0» отключается. В позициях 40,41 сообщается, что активируется резервный сетевой адаптер «eth1». В позиции 52 сообщается, что открывается сессия пользователя SWM, которая была приостановлена из-за выхода из строя сетевого адаптера «eth0».

Графики (рис. 5–7) получены по расчетам по формуле (10) для результатов отказоустойчивости (рис. 5), живучести (рис. 6) и для случая сочетания проявлений как отказоустойчивости, так и живучести (рис. 7).

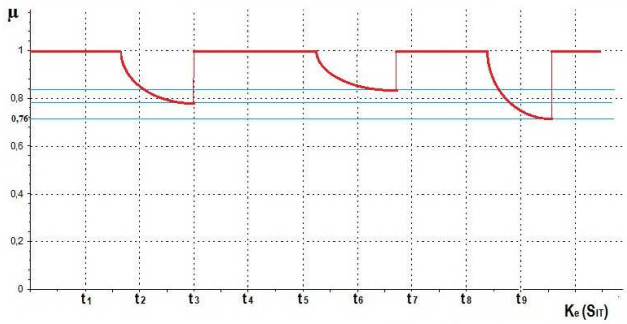


Рисунок 5 – График отказоустойчивости

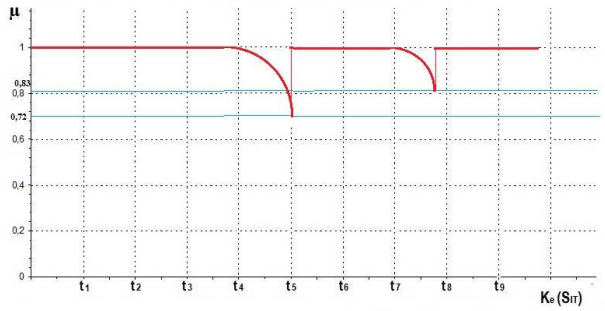


Рисунок 6 – График проявлений живучести

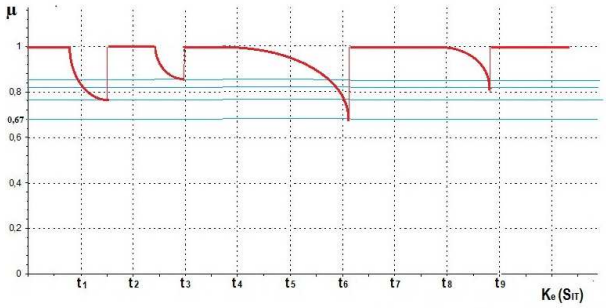


Рисунок 7 – График отражения одновременных проявлений отказоустойчивости и живучести

Результаты исследования подтверждают высокий уровень отказоустойчивости и живучести корпоративных компьютерных сетей, который составляет более 75 %.

Обсуждение и дальнейшая работа

Важным направлением дальнейших исследований для повышения эффективности ИТ является разработка метода обеспечения эффективной защиты информации непосредственно в структуре ИТ и вычислительных процессах, протекающих в процессах вычислений. Их учет в общем критерии определения эффективности ИТ позволит сбалансировать такие величины, как живучесть, отказоустойчивость и защита информации, выраженные в количественном виде, и станет основой разработки специализированной ИТ с улучшенными характеристиками.

doi.org/10.36773/1818-1212-2020-123-5-58-65

Заключение

Таким образом, разработан подход к определению эффективности ИТ на основе учета количественных величин, характеризующих отказоустойчивость и живучесть, который может быть расширен для учета других характеристических величин. Для обеспечения отказоустойчивости и живучести ИТ разработана система мероприятий, в результате выполнения которых получено ИТ узкоспециализированного использования для различных сфер применения, где сопровождения процессов относятся к системам ирреального или нереального времени с достаточно высокими параметрами отказоустойчивости, живучести и вообще резилентности и, в то же время, приемлемыми финансовыми затратами на ее эксплуатацию.

Список цитированных источников

1. Савельева, О. С. Использование индекса отказоустойчивости конструкций при проектировании / О. С. Савельева, О. М. Красножон, О. У. Лебедева. – Одесса : Одесский политехнический университет. – Труды 2. – 2014. – 130–135. doi: 10.15276/opu.2.44.2014.24.
2. Боранбаев, С. Применение метода разноплановой избыточности в облачных системах для повышения надежности / С. Боранбаев, С. Алтаев, А. Боранбаев // Труды 12-й Международной конференции по информационным технологиям : новые поколения (ITNG 2015). – Лас-Вегас, Невада, 2015. – С. 796–799.
3. Zhu, X. Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds / X. Zhu, J. Wang, H. Guo, D. Zhu, LT. Yang, L. Liu // IEEE Trans Parallel Distrib Syst 27(12) – 2016. – P. 3501–3517. [Электронный ресурс]. – Режим доступа : https://doi.org/10.1109/TPDS.2016.2543731.
4. Liu, J. Software rejuvenation based fault tolerance scheme for cloud applications In / J. Liu, J. Zhou, R. Buyya // IEEE 8th International Conference on Cloud Computing – New York, 2015. – P. 1115–1118. [Электронный ресурс]. – Режим доступа : https://doi.org/10.1109/CLOUD.2015.164.
5. Liu, J. Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability / J. Liu, S. Wang, A. Zhou, SAP Kumar, F. Yang, R. Buyya // IEEE Trans Cloud Comput PP(99):1–1. – 2016. – [Электронный ресурс]. – Режим доступа : http://dx.doi.org/10.1109/TCC.2016.2567392.
6. Nicolo, P. A frame work for self-healing software systems In // IEEE 35th International Conference on Software Engineering (ICSE), 2013. – P. 1397–1400. [Электронный ресурс]. – Режим доступа : https://doi.org/10.1109/ICSE.2013.6606726.
7. Zhao, W. Fault Tolerance Middleware for Cloud Computing In / W. Zhao, Z. Wenbing, P.M. Melliar-Smith, L.E. Moser // IEEE 3rd International Conference on Cloud Computing, 67–74. – Miami, 2010. [Электронный ресурс]. – Режим доступа : https://doi.org/10.1109/CLOUD.2010.26.
8. Bala, A. Fault tolerance- challenges, techniques and implementation in cloud computing / A. Bala, I. Chana // ISSN (Online): 16940814. IJCSI Int J Comput Sci 9(1). www.IJCSI.org., 2012.
9. Egwutuoha, I.P. A fault tolerance framework for high performance computing in cloud, Cluster, Cloud and Grid Computing (CCGrid) In / I.P. Egwutuoha, S. Chen, D. Levy, B. Selic // Proceedings of the 12th IEEE/ACM international symposium. – 13-16 May. – 2012. – P. 709–710. [Электронный ресурс]. – Режим доступа : https://doi.org/10.1109/CCGrid.2012.80.
10. Pitcher, S. New DoD Approaches on the Cyber Survivability of Weapon Systems (25 March 2019). 25 March 2019. [Online]. Available: [Электронный ресурс]. – Режим доступа : https://www.itea.org/wp-content/uploads/2019/03/Pitcher-Steve.pdf.

References

1. Savel'eva, O. S. Ispol'zovanie indeksa otkazoustojchivosti konstrukcij pri proektirovanii / O. S. Savel'eva, O. M. Krasnozhon, O. U. Lebedeva. – Odessa : Odesskij politekhnicheskij universitet. – Trudy 2. – 2014. – 130–135. doi: 10.15276/opu.2.44.2014.24.
2. Boranbaev, S. Primenenie metoda raznoplanovoj izbytochnosti v oblachnyh sistemah dlya povysheniya nadezhnosti / S. Boranbaev, S. Altaev, A. Boranbaev // Trudy 12-j Mezhdunarodnoj konferencii po informacionnym tekhnologiyam : novye pokoleniya (ITNG 2015). – Las-Vegas, Nevada, 2015. – S. 796–799.
3. Zhu, X. Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds / X. Zhu, J. Wang, H. Guo, D. Zhu, LT. Yang, L. Liu // IEEE Trans Parallel Distrib Syst 27(12) – 2016. – R. 3501–3517. [Elektronnyj resurs]. – Rezhim dostupa : https://doi.org/10.1109/TPDS.2016.2543731.
4. Liu, J. Software rejuvenation based fault tolerance scheme for cloud applications In / J. Liu, J. Zhou, R. Buyya // IEEE 8th International Conference on Cloud Computing – New York, 2015. – R. 1115–1118. [Elektronnyj resurs]. – Rezhim dostupa : https://doi.org/10.1109/CLOUD.2015.164.
5. Liu, J. Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability / J. Liu, S. Wang, A. Zhou, SAP Kumar, F. Yang, R. Buyya // IEEE Trans Cloud Comput PP(99):1–1. – 2016. – [Elektronnyj resurs]. – Rezhim dostupa : http://dx.doi.org/10.1109/TCC.2016.2567392.
6. Nicolo, P. A frame work for self-healing software systems In // IEEE 35th International Conference on Software Engineering (ICSE), 2013. – R. 1397–1400. [Elektronnyj resurs]. – Rezhim dostupa : https://doi.org/10.1109/ICSE.2013.6606726.
7. Zhao, W. Fault Tolerance Middleware for Cloud Computing In / W. Zhao, Z. Wenbing, P.M. Melliar-Smith, L.E. Moser // IEEE 3rd International Conference on Cloud Computing, 67–74. – Miami, 2010. [Elektronnyj resurs]. – Rezhim dostupa : https://doi.org/10.1109/CLOUD.2010.26.
8. Bala, A. Fault tolerance- challenges, techniques and implementa-tion in cloud computing / A. Bala, I. Chana // ISSN (Online): 16940814. IJCSI Int J Comput Sci 9(1). www.IJCSI.org., 2012.
9. Egwutuoha, I.P. A fault tolerance framework for high performance computing in cloud, Cluster, Cloud and Grid Computing (CCGrid) In / I.P. Egwutuoha, S. Chen, D. Levy, B. Selic // Proceedings of the 12th IEEE/ACM international symposium. – 13-16 May. – 2012. – P. 709–710. [Elektronnyj resurs]. – Rezhim dostupa : https://doi.org/10.1109/CCGrid.2012.80.
10. Pitcher, S. New DoD Approaches on the Cyber Survivability of Weapon Systems (25 March 2019). 25 March 2019. [Online]. Available: [Elektronnyj resurs]. – Rezhim dostupa : https://www.itea.org/wp-content/uploads/2019/03/Pitcher-Steve.pdf.

Материал поступил в редакцию 12.01.2020