УДК 004.415.2
DOI: 10.31891/2219-9365-2020-66-2-12

СТЕЦЮК М. В., КАШТАЛЬЯН А. С., ГРИБИНЧУК В. І.
Хмельницький національний університет

АРХІТЕКТУРА СПЕЦІАЛІЗОВАНИХ ІНФОРМАЦІЙНИХ СИСТЕМ З ВРАХУВАННЯМ ВИМОГ ЖИВУЧОСТІ ТА ВІДМОВОСТІЙКОСТІ В УМОВАХ ВПЛИВІВ ЗЛОВМИСНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У статті запропоновано архітектурні рішення для створення спеціалізованих інформаційних систем (ІС) з врахуванням вимог живучості та відмовостійкості в умовах впливів зловмисного програмного забезпечення, а також вимоги до апаратного забезпечення, на якому реалізуються такі архітектури ІС.

Розроблена архітектура та вимоги до її реалізації та апаратної підтримки, які враховують відмовостійкість та живучість, та можуть бути розширені для врахування інших характеристичних величин. Для забезпечення відмовостійкості та живучості ІС розроблено систему заходів в результаті виконання яких отримано ІС вузькоспеціалізованого використання для різних сфер застосування. Живучість ІС забезпечується: резервуванням серверної частини з територіальним рознесенням основного і резервного сервера, особливістю резервування є те, що функцію сервера, в критичний момент, переймає на себе дзеркальний SQL-сервер, який в штатному режимі забезпечує роботу FTP-сервера; резервуванням програмного забезпечення клієнтської частини, особливістю резервування є те, що в якості резерву слугує не спеціально виділений комп'ютерів, а резерв продуктивності окремих клієнтських комп'ютерів, на які, згідно плану резервування, встановлюється програмного забезпечення клієнтської частини, що резервуються, яке в критичний момент буде використовуватись як штатне, не допускаючи втрати функціональності ІС.

Проведені експериментальні дослідження підтверджують можливість застосування запропонованих рішень стосовно архітектури ІС.

У результаті використання розроблених заходів було отримано архітектуру ІС вузькоспеціалізованого використання для різних сферах застосування, де супроводжуванні процеси відносяться до ірреального або нереального часу із досить високими параметрами відмовостійкості, живучості та загалом резилентності.

Ключові слова: інформаційні системи, архітектура, відмовостійкість, живучість, комп'ютерні мережі.

STETSYUK M., KASHTALIAN A., GRIBINCHOOK V.
Khmelnytsky National University

ARCHITECTURE OF SPECIALIZED INFORMATION SYSTEMS, TAKING INTO ACCOUNT THE REQUIREMENTS OF RESISTANCE AND RESISTANCE FROM FAILURE IN THE CONDITIONS OF THE EFFECTS OF MALWARE SOFTWARE

The article proposes architectural solutions for the creation of specialized information systems (IS) taking into account the requirements of survivability and fault tolerance in the conditions of malicious software, as well as the requirements for the hardware on which such IP architectures are implemented.

Developed architecture and requirements for its implementation and hardware support, which take into account fault tolerance and survivability, and can be extended to take into account other characteristics. To ensure fault tolerance and survivability of IP, a system of measures has been developed, as a result of which IPs of narrowly specialized use for different areas of application have been obtained. The survivability of the IS is provided by: redundancy of the server part with the territorial separation of the main and backup server, the feature of redundancy is that the server function, at a critical moment, takes over the mirror SQL-server, which in normal mode provides FTP-server; reservation of the software of the client part, the peculiarity of the reservation is that the reserve is not a dedicated computer, and the performance reserve of individual client computers, which, according to the backup plan, is installed software of the client part of the reserved, which in the critical moment will be used as a regular, preventing the loss of IP functionality.

The conducted experimental researches confirm the possibility of application of the offered decisions concerning IP architecture.

As a result of using the developed measures, the architecture of IP of specialized use for different areas of application was obtained, where the accompanying processes belong to unreal or unreal time with rather high parameters of fault tolerance, survivability and overall resistance.

Keywords: information systems, architecture, fault tolerance, survivability, computer networks.

Вступ. Постановка задачі дослідження. Відмовостійкість та живучість [1] забезпечують ефективність ІС, яка досягається різними шляхами. Одним із її параметрів є час недоступності, тобто час, коли система не в змозі виконувати свої функції в рамках вимог до неї. Для різних систем цей час різний і знаходиться в діапазоні від нуля до певної порогової прийнятної величини. Для спеціалізованих ІС, які функціонують в корпоративних комп'ютерних мережах та виконують функцію інформаційного забезпечення в такій вузькоспеціалізованій предметній області, як фінансово-господарська діяльність в різних сферах застосування, цей параметр значно вище нуля, але вимоги до таких ІС теж достатньо високі, особливо при постійному зростанні їх кількісних параметрів функціонування (збільшення числа користувачів, складності інформаційних потоків та об'ємів оброблюваних даних) та роботи в умовах впливів зловмисного ПЗ.

Живучість розробленої ІС забезпечується: резервуванням серверної частини ІС з територіальним рознесенням основного і резервного сервера, особливістю резервування є те, що функцію сервера, в критичний момент, переймає на себе дзеркальний SQL-сервер, який в штатному режимі забезпечує роботу FTP-сервера; резервуванням програмного забезпечення клієнтської частини, особливістю якого є те, що в якості резерву слугує не спеціально виділений комп'ютер, а резерв продуктивності окремих клієнтських комп'ютерів, на які, згідно плану резервування, встановлюється ПЗ клієнтської частини, що резервуються, яке в критичний момент буде використовуватись як штатне, не допускаючи втрати функціональності ІС.

Відмовостійкість її клієнтської частини ІС забезпечено шляхом виконання комплексу заходів, що включає окрім традиційно апаратної надмірності ще і функціональну надмірність: організація автоматичного оновлення системного та прикладного ПЗ клієнтських ПК шляхом моніторингу його актуальності із заданою періодичністю; алгоритми процедур, які реалізують критичні функції клієнтської частини ІС, із включенням в них нетривіального (інтелектуального) блоку обробки помилок, що виконується паралельно із самою процедурою; використання нетривіальних редакторів даних, які включають до свого алгоритму роботи інтерактивну процедуру, що виключає неконрольоване маніпулювання даними бази даних, оператором; реалізацією критичних по використанню ресурсів, розрахункових процедур із можливістю оперативного вибору місця їх виконання, що не допускає перевантаження засобів апаратної платформи для ІС. В цих умовах актуальності набирає питання розробки таких архітектур спеціалізованих ІС, які б могли підтримувати свою живучість та відмовостійкість в умовах впливу зловмисного програмного забезпечення. Це було б одним із елементів з покращення рівня безпеки в корпоративних комп'ютерних мережах. І його розвиток дозволив би створювати функціонально живучіші та відмовостійкіші спеціалізовані ІС. Відомі архітектури недостатньо забезпечують рівні живучості та відмовостійкості, тому метою роботи є подальший розвиток відомих архітектур і створення нових для покращення безпеки в корпоративних комп'ютерних мережах за рахунок підвищення рівнів живучості та стійкості безпосередньо спеціалізованих ІС.

1. Відомі методи створення спеціалізованих ІС на основі врахування критеріїв живучості та відмовостійкості в умовах впливу зловмисного програмного забезпечення

Відмовостійкістю ІС вважатимемо властивість системи зберігати повну або часткову працездатність у випадках відмов окремих елементів, що непов'язані із зовнішніми нерегламентованими діями. Під живучістю інформаційної системи розумітимемо її властивість залишатися працездатною з допустимим зменшенням продуктивності в умовах негативних зовнішніх впливів [1]. Ці поняття визначають таку мету як забезпечення доступності ІТ, що досягається різними шляхами. Від забезпечення цих параметрів в прямій залежності знаходиться ефективність функціонування всієї спеціалізованої ІС.

Інформаційну технологію оцінки надійності технічних об'єктів, структура якої відповідає одному з відомих типів нейронних мереж запропоновано в роботі [2]. В роботах [3, 4] розглянуто хмарні програми як складові з декількох компонентів хмарних служб, що спілкуються між собою через інтерфейси веб-служб. В роботах [5, 6] розроблено забезпечення надійності інформаційних систем. Для підвищення надійності інформаційних систем використано метод, базований на оцінці та нейтралізації ризиків. В роботі [7] показано як налагоджувані програмні системи складаються з великої кількості різних, критичних, некритичних та взаємозалежних конфігурацій. В [8] розглянуто як хмари стають важливою платформою для наукової роботи за рахунок програм. В роботі [9] представлено метод та технологія для уникнення функціональних збоїв під час виконання в прикладних системах. В [10] запропонована проактивна схема відновлення, заснована на міграції служб. В роботі [11] толерантність відмов є головним питанням гарантування доступності та надійності критичних послуг. В роботі [12] показано як хмарні обчислення пропонують нову потужність та гнучкість для високоефективних обчислювальних програм із забезпеченням великої кількості віртуальних машин для обчислювальних програм. В роботах [13–16] кібер-стійкість та кібер-життєздатність представлено як тісно пов'язані між собою поняття. В роботах [17–24] показано вплив на відмовостійкість та живучість ІС комп'ютерних атак та різних типів зловмисного програмного забезпечення.

Відомі архітектури базовані на критеріях забезпечення відмовостійкості та живучості спеціалізованих ІС недостатньо систематизовані та не завжди можуть бути реалізовані через специфіку використання. Тому, необхідним є подальше дослідження та розробка нових архітектур ІС, які б дозволили покращити їх відмовостійкість та живучість, зокрема і від кібер-атак та зловмисного програмного забезпечення.

2. Забезпечення живучості та відмовостійкості спеціалізованих ІС в умовах руйнуючого впливу зловмисного програмного забезпечення за рахунок удосконалення архітектур систем

Основними напрямами для підвищення живучості та відмовостійкості ІС є внесення надмірності в конфігурацію апаратних і програмних засобів, підтримуючої інфраструктури, резервування інформаційних ресурсів (програм та даних). При цьому ІС повинна відповідати наступним основним вимогам: система повинна будуватись так, щоб в ній був відсутній компонент (ресурс), відмова якого призведе до повної відмови всієї системи. Для систем реального часу, додатково, накладаються часові обмеження досягнення результату. Розглядатимемо спеціалізовану ІС, яка відноситься до систем ірреального часу. Тому часові обмеження для неї набагато менш жорсткі, порівняно з системами реального часу. В зв'язку з вибраною для

розгляду архітектурою реалізації ІС, яка складається із серверної та клієнтської частин, то відповідно розглядатимемо питання відмовостійкості та живучості в співвіднесенні до функцій покладених на них.

Головною властивістю відмовостійкості є прозорість відмов її окремих компонентів для кінцевого користувача. Це означає, що відмовостійка система автоматично змінює свою конфігурацію у випадку відмови. Її програмне забезпечення в процесі виконання шукає обхідні шляхи, намагаючись в умовах відмови, привести виконувану функцію до успішного завершення. Для ІС, призначених для інформаційного забезпечення у вузькій спеціалізованій предметній області, наприклад фінансово-господарська діяльність закладу вищої освіти, буде доцільним відмовитись від автоматичної системи керування відмовостійкістю на користь автоматизованої. При такому підході частина дорогих функцій управління надмірностями, присутніми в ІС, буде покладена на людину, якщо це не загрожує можливими значними втратами. Але вирішенням задачі побудови ІС, є не забезпечення максимально можливої відмовостійкості системи, а знаходження прийнятного балансу параметрів системи, в рамках певного технологічного базису. А, також, в тому числі враховуючи вимоги критерію «відмовостійкість \ вартість». Дослідимо вирішення питань забезпечення відмовостійкості ІС при використанні такої стратегії. Проаналізуємо фактори, що негативно впливають на відмовостійкість ІС зі сторони клієнта. Це потрібно для того, щоб оцінити і виробити адекватні заходи протидії. Схему впливу негативних факторів на відмостійкість клієнтської частини спеціалізованої ІС зображено на рис. 1.

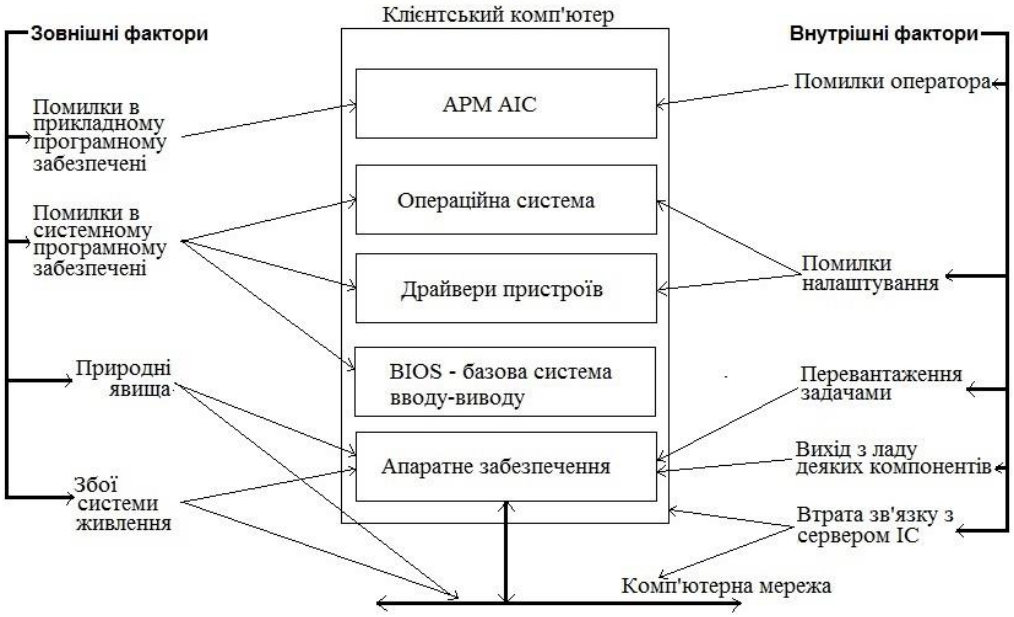
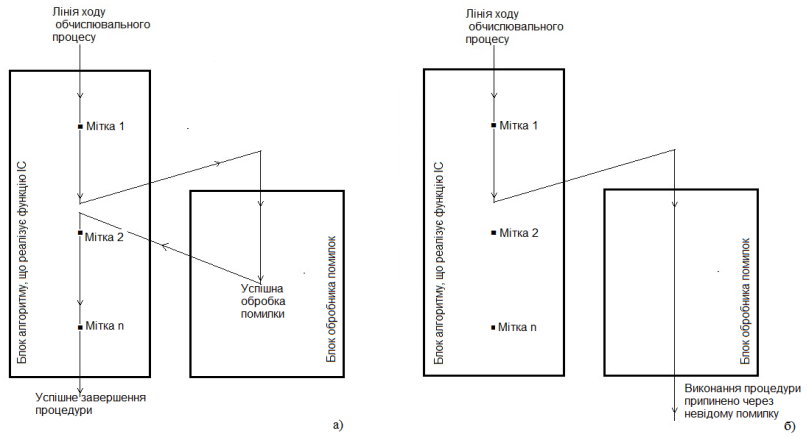


Рис. 1. Схема дії негативних факторів, що впливають на відмовостійкість клієнтської частини спеціалізованої ІС

Негативні фактори, що впливають на відмовостійкість клієнтської частини ІС поділено на зовнішні та внутрішні. Серед зовнішніх факторів найбільшу загрозу представляють собою збої в роботі енергосистем живлення та природні явища, які можуть призвести до відмов компонентів комп'ютерів та комп'ютерних мереж. Іншим важливим фактором є помилки в коді системного програмного забезпечення. Зменшити його прояви можна шляхом використання програмного забезпечення з автоматичним налаштуванням, що не завжди прийнятно, та залученням більш кваліфікованого персоналу.

Для прикладного програмного забезпечення, до якого відносяться клієнтські частини спеціалізованої ІС, то критичні помилки, які проявились в ході експлуатації робочих місць, фіксуються разом із своїми параметрами в реєстрі системи в автоматичний спосіб і, в подальшому використовується для аналізу з метою усунення причин, що їх викликали. Це досягається завдяки підходу, який базований на привнесенні деякої надмірності в програмне забезпечення клієнтської частини ІС. З цією метою всі розрахункові процедури, які можуть містити критичні для функціонування помилки, розроблені з дотриманням певного однотипного шаблону побудови алгоритму її виконання. Суть алгоритму відображена на рис. 2.

В цій структурі алгоритм виконання будь-якої нетривіальної процедури розділяється на два взаємодіючих блоки. В першому блоці реалізується функція процедури ІС, а в другому обробник помилок. В процесі виконання деякої процедури, яка реалізує одну із функцій компоненти ІС, обидва блоки взаємодіють між собою, передаючи управління обчислювальним процесом один одному, поки виконувана функція не завершиться. Його суть полягає в тому, що алгоритм, який реалізує функцію ІС, розділяється маркерами (мітка 1,..., мітка *n* на рис. 2) на фрагменти за принципом функціональної завершеності.



Мал. 2. Модель роботи алгоритма відмовостійкої процедури. а) для випадку успішного завершення процедури після виникнення помилки; б) для випадку, коли помилка., невідома для обробника помилок.

Рис. 2. Модель роботи алгоритму відвостійкої процедури:
***а*) для випадку успішного завершення процедури; *б*) для випадку, коли помилка, невідома для обробника помилок**

Перед початком виконання поточного фрагменту алгоритму в реєстр фатальних помилок заноситься інформація про гіпотетично можливу помилку (код екземпляра робочого місця клієнта, код функції, номер мітки, час тощо). У подальшому можливі наступні варіанти розвитку подій:

1. Фрагмент алгоритму функції успішно виконався. В цьому випадку інформація в реєстрі про помилку, що не сталась, знищується, а обчислювальний процес переходить до виконання наступного фрагмента.

2. У процесі виконання фрагменту сталась помилка, але вона успішно локалізована обробником помилок (рис. 2, *а*). В цьому випадку інформація про помилку також може бути видалена з реєстру.

3. У процесі виконання фрагмента сталась помилка, яка не була локалізована обробником помилок (рис. 2, *б*). В цьому випадку інформація про можливу помилку залишиться в реєстрі.

Більш детальніше алгоритм виконання нетривіальної процедури із застосуванням вище згаданого шаблону зображено на рис. 3. Такий підхід до побудови алгоритму виконання функції ІС дозволяє фіксувати в реєстрі помилок, також, і інформацію про помилки, викликані збоями або критичними відмовами в апаратному забезпечені клієнтського комп'ютера, його системного програмного забезпечення.

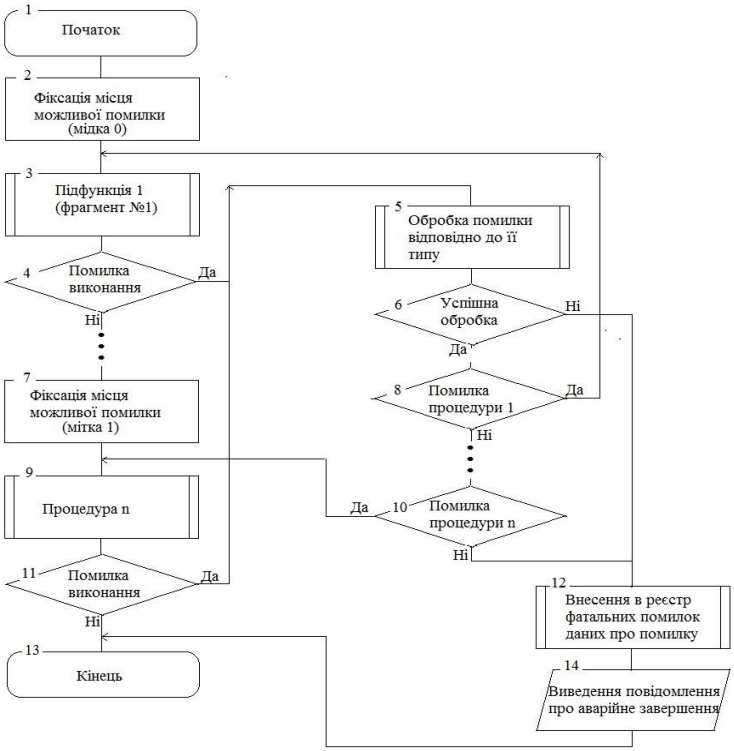


Рис. 3. Алгоритм обробки помилок в підсистемі забезпечення відмовостійкості

Зібрана в такий спосіб інформація про фатальні помилки, що стались в процесі функціонування ІС, дозволяє їх класифікувати та, в процесі подальшого аналізу виявити слабкі ланки з метою їх усунення, шляхом удосконалення програмного забезпечення клієнтської частини.

Розглянемо внутрішні фактори, що впливають на відмовостійкість клієнтської частини ІТ (рис. 1) та методи, які були застосовані з метою його зменшення. Перший з них, за частотою виникнення, відбувається через помилки оператора. Ця проблема вирішена шляхом використання типового редактора даних, в якому всі процедури внесення змін в базу даних реалізовані з використанням шаблона (рис. 4), структура якого включає надмірності у вигляді блоків алгоритму, які передбачають перевірку дій оператора.

Рис. 4. Алгоритм шаблону реалізації функцій редагування баз даних

Як видно з блок-схеми шаблона типового редактора (рис. 4), оператор не має змоги прямого редагування даних. Всі його дії по маніпулюванню даними знаходяться під контролем процедур попередньої перевірки, які включають в себе набір правил (допустима повнота внесення даних, знаходження значень в заданих діапазонах, відсутність протиріч з раніше внесеними даними і т.д.), контекстно зв'язаних із виконуваною функцією. Таким чином, фактор помилок оператора знімається, шляхом ускладнення алгоритму роботи редактора даних.

Наступним із значимих внутрішніх факторів, що негативно впливають на відмостійкість є перевантаження апаратної платформи клієнтського комп'ютера задачами, що може різко погіршити часові параметри виконуваних завдань клієнтською частиною ІС, або навіть зробити неможливою його роботу, через вичерпання технічних ресурсів. Щоб нейтралізувати дію цього фактора, при розробці програмного забезпечення, а саме тієї його частини, яка відповідальна за реалізацію «бізнес-логіки» застосоване функціональне резервування (рис. 5).

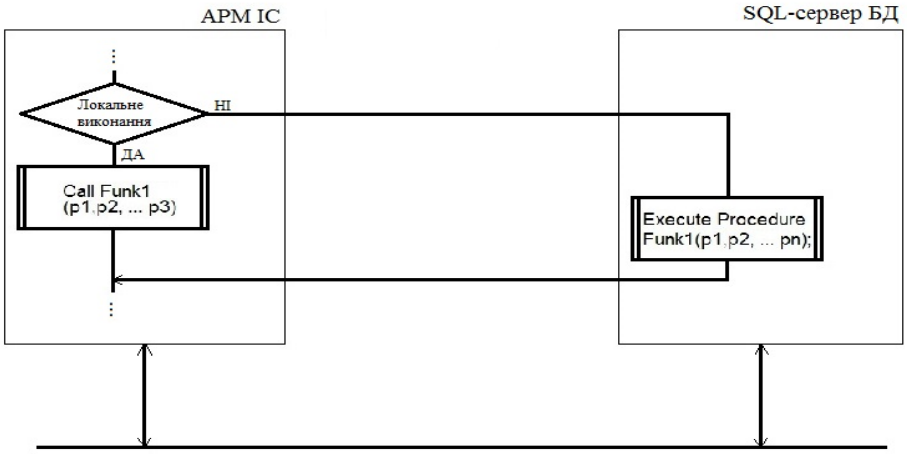


Рис. 5. Застосування функціонального резервування функцій ІС

Наявність функціонального резерву «важких» розрахункових функцій дозволяє здійснювати маневр обчислювальними потужностями апаратної платформи, в разі перевантаження окремих її ланок, підвищуючи таким чином відмовостійкість.

Таким чином, відмовостійкість клієнтської частини забезпечено шляхом виконання комплексу заходів, що включає апаратну та функціональну надмірності: живлення клієнтських ПК від окремої лінії з пристроями захисту; використання пристроїв грозозахисту в лініях комп'ютерної мережі; організація автоматичного оновлення системного програмного забезпечення клієнтських ПК; розробка алгоритмів процедур, які реалізують критичні функції клієнтської частини ІС, із включенням в них нетривіального (інтелектуального) блоку обробки помилок, що виконується паралельно із самою процедурою; використання нетривіальних редакторів даних, які включають до свого алгоритму роботи інтерактивну процедуру, що виключає неконтрольоване маніпулювання даними бази даних, оператором; реалізацією критичних по використанню ресурсів, розрахункових процедур із можливістю оперативного вибору місця їх виконання, що не допускає перевантаження апаратних засобів. Отже, процес забезпечення відмовостійкості є неперервним протягом всього життєвого циклу ІС. Він розпочинається з планування заходів забезпечення відмовостійкості, що проектується і триває до часу завершення її функціонування взагалі.

Показники живучості в складній системі: багатофункціональність окремих компонент; наявність єдиної (головної) мети функціонування всієї системи; можливість не тільки інформаційного обміну між окремими компонентами, але й інформаційної взаємодії з користувачами; наявність засобів захисту, контролю, діагностики й самоорганізації. Задача аналізу структурної живучості потребує визначення: системної архітектури, необхідної для виконання цілі функці-онування ІС у деякий момент або проміжок часу, коли виникають небажані впливи на систему; вимог щодо окремих видів ресурсів системи та їх взаємозв'язку; вимог щодо функціональних можливостей ресурсів системи; особливостей характеру небажаних впливів чи їх наслідків. З рис. 6 видно, як в рамках розробленої системи реалізована задача підвищення живучості ІС, шляхом структурного резервування основних її компонентів, а саме серверної її частини. У випадку виходу з ладу основного сервера ІС, його функції може взяти на себе резервний, який має абсолютно однакові налаштування з основним. При цьому основний та резервний сервери рознесені територіально і живляться з різних ліній. Оскільки вихід з ладу зразу двох серверів є подія малоймовірна, то цим забезпечується висока живучість серверної частини ІС. Реконфігурація реальної системи займає не більше 10 хв. Копія бази даних підтримується в актуальному стані службою реплікацій, тому заміна основної бази даних на базу даних – копію виконується без втрат інформації. Але незначна втрата інформації при такій схемі все ж можлива. Це може трапитись при відмові деяких чутливих компонентів апаратної платформи сервера. Як правило, це останні запущені транзакції, виконання яких буде припинене через відмову обладнання. І якщо це транзакції на зміну інформації в базі даних, то в цьому випадку інформація буде втрачена. Але оскільки така подія в життєвому циклі інформаційної системи сама по собі рідкісна, то такою можливою кількістю втрати інформації можна знехтувати. Після відновлення роботи серверної частини, операторам, чиї транзакції були втрачені, потрібно повторно виконати останні операції, для відновлення втраченої інформації. При регулярному діагностуванні критичного обладнання сервера можна в більшості випадків виявити назріваючу відмову і вчасно замінити відповідний компонент. Так організація роботи дозволяє зменшити вірогідність виходу з ладу серверної частини ІС і тим самим звести нанівець втрату інформації.

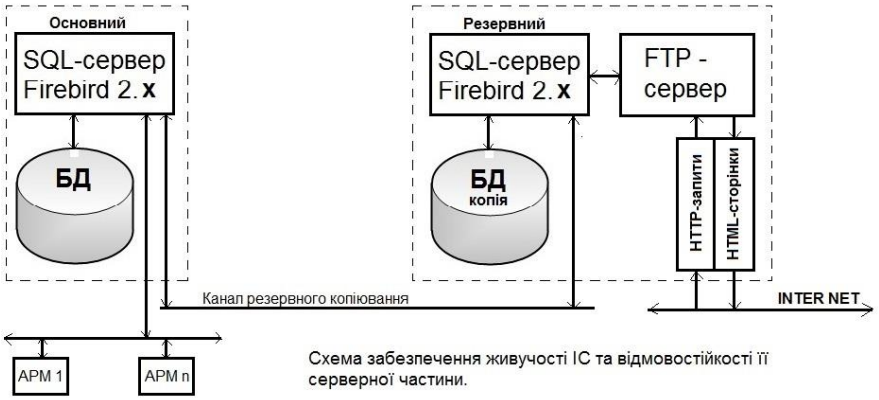


Рис. 6. Схема резервування серверної частини ІС

Таким чином, живучість ІС забезпечується: резервуванням серверної частини ІС з територіальним рознесенням основного і резервного сервера, особливістю резервування є те, що функцію сервера, в критичний момент, переймає на себе дзеркальний SQL-сервер, який в штатному режимі забезпечує роботу

FTP-сервера; резервуванням програмного забезпечення клієнтської частини, особливістю резервування є те, що в якості резерву слугує не спеціально виділений комп'ютерів, а резерв продуктивності окремих клієнтських комп'ютерів, на які, згідно плану резервування, встановлюється програмного забезпечення клієнтської частини, що резервуються, яке в критичний момент буде використовуватись як штатне, не допускаючи втрати функціональності ІС. В результаті використання перелічених заходів було отримано архітектуру ІС вузькоспеціалізованого використання для різних сферах застосування, де супроводжуванні процеси відносяться до ірреального або нереального часу із досить високими параметрами відмовостійкості, живучості та загалом резилентності.

3. Експерименти

Для визначення на скільки ефективними є запропоновані рішення із забезпечення відмовостійкості та живучості проведемо порівняння критерію ефективності для ІТ без забезпечення відмовостійкості та живучості і з включенням цих характеристик на основі формули (1):

$$K_e(S_{IT}) = \alpha_{1,j,p,q} \cdot \frac{T_{f_1(s_i),1} - (T_{f_1(s_i),2} + T_{f_1(s_i),3})}{T_{f_1(s_i),1}} + \alpha_{2,j,p,q} \cdot \frac{T_{f_2(s_i),1} + T_{f_2(s_i),2}}{T_{f_2(s_i),1}}, \quad (1)$$

де $\alpha_{1,j,p,q}$ – коефіцієнт для значення, яке визначає відмовостійкість в кількісних одиницях; $\alpha_{2,j,p,q}$ – коефіцієнт для значення, яке визначає живучість в кількісних одиницях; $\alpha_{1,j,p,q} + \alpha_{2,j,p,q} = 1$.

Значення величини критерія ефективності ІТ, в якій не забезпечуються вимоги відмовостійкості і живучості отримуємо з формули (1) так: 1) вирішення проблем, пов'язаних із відсутністю забезпечення в ІТ реалізованих відмовостійкості та живучості, покладено на оператора чи адміністратора, який постійно моніторить функціонування ІТ; вирішення проблемних ситуацій здійснюється тільки при їх виявленні. Ефективним значенням є значення мінімально відхилене від одиниці. Результати забезпечення відмовостійкості та живучості спеціалізованої ІТ зображені в реалізованій ІТ на рис. 7.

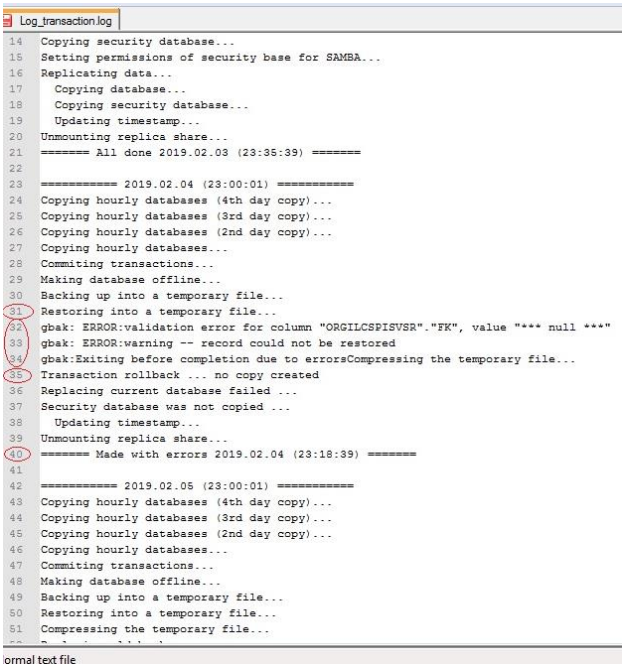


Рис. 7. Фрагмент лог-файла підсистеми резервного копіювання бази даних

Цей фрагмент відображає роботу підсистеми транзакцій бази даних під час її резервного копіювання. При виконані процедури створення резервної копії бази даних утилітою GBAK 4 лютого 2019 року сталась помилка в даних, що призвела до відкату транзакції. Критичні позиції виділені.

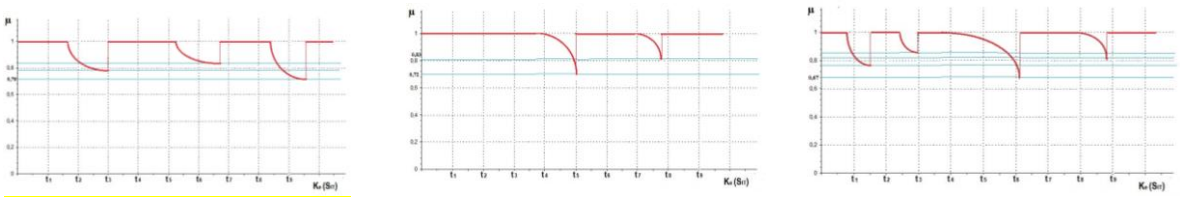
Позиція 31. Створення тимчасового файла бази даних.

Позиція 32–34. Повідомлення утиліти GBAK про помилку при спробі запису в поле [FK] таблиці ORGILCSPISVSR значення по умовчанню визначника NULL.

Позиція 35. Відкат поточної транзакції через помилку.

Позиція 40. Сповіщення, що процедура створення резервної копії завершилась із помилками.

Результати відмовостійкості, графік проявів живучості та графік відображення одночасних проявів та відмовостійкості та живучості зображені на рис. 8.



а *б* *в*

Рис. 8. Графік відмовостійкості (*а*); графік проявів живучості (*б*);
графік відображення одночасних проявів та відмовостійкості та живучості (*в*)

Результати дослідження підтверджують високий рівень стійкості та виживання в корпоративних комп'ютерних мережах, який становить понад 75%.

Напрями подальших досліджень. Важливим напрямом подальших досліджень для покращення ефективності ІС є розробка методу забезпечення ефективного захисту інформації в архітектурі ІС. Їх врахування в загальному критерії визначенні ефективності ІС дозволить збалансувати такі величини як живучість, відмовостійкість та захист інформації, виражені в кількісному вигляді, та стане основою розробки спеціалізованої ІС з покращеними характеристиками.

Висновки. Таким чином, розроблена архітектура та вимоги до її реалізації та апаратної підтримки, які враховують відмовостійкість та живучість, та можуть бути розширені для врахування інших характеристичних величин. Для забезпечення відмовостійкості та живучості ІС розроблено систему заходів в результаті виконання яких отримано ІС вузькоспеціалізованого використання для різних сфер застосування, де супроводжуванні процеси відносяться до ірреального або нереального часу із досить високими параметрами відмовостійкості, живучості та загалом резилентності і, в той же час, прийнятним рівним фінансових витрат на її експлуатацію. Проведені експериментальні дослідження підтверджують можливість застосування запропонованих рішень стосовно архітектури ІС.

Література

1. ДСТУ 3396.2-97 Protection of information. Technical protection of information. Terms and definitions. State Committee of Ukraine, Kyiv (1997) [in Ukrainian]

2. Savelyeva, O. S., Krasnozhon, O. M., Lebedeva, O. U. (2014). Using the structural fault-tolerance index in project designing. Odes'kyi Politechnichnyi Universytet. Pratsi, 2, 130–135. doi: 10.15276/opu.2.44.2014.24.

3. Liu J, Zhou J, Buyya R (2015) Software rejuvenation based fault tolerance scheme for cloud applications In: 2015 IEEE 8th International Conference on Cloud Computing, 1115–1118, New York. https://doi.org/10.1109/CLOUD.2015.164.

4. Liu J, Wang S, Zhou A, Kumar SAP, Yang F, Buyya R (2016) Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability. IEEE Trans Cloud Comput PP(99):1–1. http://dx.doi.org/10.1109/TCC.2016.2567392.

5. S. Boranbayev, S. Altayev, A. Boranbayev. Applying the method of diverse redundancy in cloud based systems for increasing reliability, in Proceedings of the 12th International Conference on Information Technology: New Generations (ITNG 2015) (Las Vegas, Nevada, 2015), pp. 796–799.

6. Boranbayev A., Boranbayev S., Yersakhanov K., Nurusheva A., Taberkhan R. (2018) Methods of Ensuring the Reliability and Fault Tolerance of Information Systems. In: Latifi S. (eds) Information Technology - New Generations. Advances in Intelligent Systems and Computing, vol 738. Springer, Cham.

7. Chinnaiah, M., Niranjan, N. Fault tolerant software systems using software configurations for cloud computing. J Cloud Comp 7, 3 (2018). https://doi.org/10.1186/s13677-018-0104-9.

8. Zhu X, Wang J, Guo H, Zhu D, Yang LT, Liu L (2016) Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. IEEE Trans Parallel Distrib Syst 27(12):3501–3517. https://doi.org/10.1109/TPDS.2016.2543731.

9. Nicolo P (2013) A frame work for self-healing software systems In: IEEE 35th International Conference on Software Engineering (ICSE), 1397–1400. https://doi.org/10.1109/ICSE.2013.6606726.

10. Zhao W, Wenbing Z, Melliar-Smith PM, Moser LE (2010) Fault Tolerance Middleware for Cloud Computing In: 2010 IEEE 3rd International Conference on Cloud Computing, 67–74, Miami. https://doi.org/10.1109/CLOUD.2010.26.

11. Bala A, Chana I (2012) Fault tolerance- challenges, techniques and implementation in cloud computing, ISSN (Online): 16940814. IJCSI Int J Comput Sci 9(1). www.IJCSI.org.

12. Egwutuoha IP, Chen S, Levy D, Selic B (2012) A fault tolerance framework for high performance computing in cloud, Cluster, Cloud and Grid Computing (CCGrid) In: Proceedings of the 12th IEEE/ACM international symposium. 13-16 May, 709–710. https://doi.org/10.1109/CCGrid.2012.80.

13. S. Pitcher, "New DoD Approaches on the Cyber Survivability of Weapon Systems (25 March 2019)," 25 March 2019. [Online]. Available: https://www.itea.org/wp-content/uploads/2019/03/Pitcher-Steve.pdf.

14. D. J. Bodeau, R. D. Graubart, R. M. McQuaid and J. Woodill, "Cyber Resiliency Metrics and Scoring in Practice: Use Case Methodology and Examples (MTR 180449)," The MITRE Corporation, Bedford, MA, 2018.

15. D. Fitzpatrick, D. Bodeau, R. Graubart, R. McQuaid, C. Olin and J. Woodill, "(DRAFT) Cyber Resiliency Evaluation Framework for Weapon Systems: Foundational Principles and Their Potential Effects on Adversaries," The MITRE Corporation, Bedford, MA, 2019.

16. NIST, "Initial Public Draft of NIST SSP 800-160 Volume 2, Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems," 21 March 2018. [Online]. Available: https://csrc.nist.gov/CSRC/media/Publications/sp/800-160/vol-2/draft/documents/sp800-160-vol2-draft.pdf.

17. Lysenko, S., Savenko, O., Kryshchuk, A., Kljots, Y. Botnet detection technique for corporate area network. In: Proc. of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, pp. 363-368 (2013).

18. Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search. CEUR Workshop, Vol. 1844, pp. 555–569 (2017).

19. Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Approach for the Unknown Metamorphic Virus Detection. In: 9-th IEEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems. Technology and Applications, pp. 453–458 (2017).

20. Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A. and Nicheporuk, A.: A technique for detection of bots which are using polymorphic code. In: 21st International Conference, CN, Springer, Brunów, Poland, pp. 265-276 (2014)

21. Kondratenko, Y., Kondratenko, N.: Soft Computing Analytic Models for Increasing Efficiency of Fuzzy Information Processing in Decision Support Systems. Chapter in book: Decision Making: Processes, Behavioral Influences and Role in Business Management, R. Hudson (Ed.), Nova Science Publishers, New York, 41-78 (2015)

22. Savenko O.S Research of methods of antiviral diagnostics of computer networks / O.S Savenko, S.M Lysenko // Visnyk of Khmelnytsky National University. Technical sciences. - 2007. - № 2, v. 2. - P. 120–126.(in Ukrainian)

23. Savenko O.S., Payuk V.P., Savenko B.O, Kashtalyan A.S. Models of undocumented software bookmarks in local computer networks / Measuring and computing equipment in technological processes / 2019. - №2. - P.84-90.(in Ukranian)

24. Савенко О.С. Модель процесу пошуку троянських програм в персональному комп'ютері / О.С. Савенко, С.М. Лисенко //Радіоелектронні і комп'ютерні системи. – 2008. – №7. – С.87-92.

References

1. DSTU 3396.2-97 Protection of information. Technical protection of information. Terms and definitions. State Committee of Ukraine, Kyiv (1997) [in Ukrainian]

2. Savelyeva, O. S., Krasnozhon, O. M., Lebedeva, O. U. (2014). Using the structural fault-tolerance index in project designing. Odes'kyi Politechnichnyi Universytet. Pratsi, 2, 130–135. doi: 10.15276/opu.2.44.2014.24.

3. Liu J, Zhou J, Buyya R (2015) Software rejuvenation based fault tolerance scheme for cloud applications In: 2015 IEEE 8th International Conference on Cloud Computing, 1115–1118, New York. https://doi.org/10.1109/CLOUD.2015.164.

4. Liu J, Wang S, Zhou A, Kumar SAP, Yang F, Buyya R (2016) Using Proactive Fault-Tolerance Approach to Enhance Cloud Service Reliability. IEEE Trans Cloud Comput PP(99):1–1. http://dx.doi.org/10.1109/TCC.2016.2567392.

5. S. Boranbayev, S. Altayev, A. Boranbayev. Applying the method of diverse redundancy in cloud based systems for increasing reliability, in Proceedings of the 12th International Conference on Information Technology: New Generations (ITNG 2015) (Las Vegas, Nevada, 2015), pp. 796–799.

6. Boranbayev A., Boranbayev S., Yersakhanov K., Nurusheva A., Taberkhan R. (2018) Methods of Ensuring the Reliability and Fault Tolerance of Information Systems. In: Latifi S. (eds) Information Technology - New Generations. Advances in Intelligent Systems and Computing, vol 738. Springer, Cham.

7. Chinnaiah, M., Niranjan, N. Fault tolerant software systems using software configurations for cloud computing. J Cloud Comp 7, 3 (2018). https://doi.org/10.1186/s13677-018-0104-9.

8. Zhu X, Wang J, Guo H, Zhu D, Yang LT, Liu L (2016) Fault-tolerant scheduling for real-time scientific workflows with elastic resource provisioning in virtualized clouds. IEEE Trans Parallel Distrib Syst 27(12):3501–3517. https://doi.org/10.1109/TPDS.2016.2543731.

9. Nicolo P (2013) A frame work for self-healing software systems In: IEEE 35th International Conference on Software Engineering (ICSE), 1397–1400. https://doi.org/10.1109/ICSE.2013.6606726.

10. Zhao W, Wenbing Z, Melliar-Smith PM, Moser LE (2010) Fault Tolerance Middleware for Cloud Computing In: 2010 IEEE 3rd International Conference on Cloud Computing, 67–74, Miami. https://doi.org/10.1109/CLOUD.2010.26.

11. Bala A, Chana I (2012) Fault tolerance- challenges, techniques and implementation in cloud computing, ISSN (Online): 16940814. IJCSI Int J Comput Sci 9(1). www.IJCSI.org.

12. Egwutuoha IP, Chen S, Levy D, Selic B (2012) A fault tolerance framework for high performance computing in cloud, Cluster, Cloud and Grid Computing (CCGrid) In: Proceedings of the 12th IEEE/ACM international symposium. 13-16 May, 709–710. https://doi.org/10.1109/CCGrid.2012.80.

13. S. Pitcher, "New DoD Approaches on the Cyber Survivability of Weapon Systems (25 March 2019)," 25 March 2019. [Online]. Available: https://www.itea.org/wp-content/uploads/2019/03/Pitcher-Steve.pdf.

14. D. J. Bodeau, R. D. Graubart, R. M. McQuaid and J. Woodill, "Cyber Resiliency Metrics and Scoring in Practice: Use Case Methodology and Examples (MTR 180449)," The MITRE Corporation, Bedford, MA, 2018.

15. D. Fitzpatrick, D. Bodeau, R. Graubart, R. McQuaid, C. Olin and J. Woodill, "(DRAFT) Cyber Resiliency Evaluation Framework for Weapon Systems: Foundational Principles and Their Potential Effects on Adversaries," The MITRE Corporation, Bedford, MA, 2019.

16. NIST, "Initial Public Draft of NIST SSP 800-160 Volume 2, Systems Security Engineering: Cyber Resiliency Considerations for the Engineering of Trustworthy Secure Systems," 21 March 2018. [Online]. Available: https://csrc.nist.gov/CSRC/media/Publications/sp/800-160/vol-2/draft/documents/sp800-160-vol2-draft.pdf.

17. Lysenko, S., Savenko, O., Kryshchuk, A., Kljots, Y. Botnet detection technique for corporate area network. In: Proc. of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, pp. 363-368 (2013).

18. Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Metamorphic Viruses' Detection Technique Based on the Equivalent Functional Block Search. CEUR Workshop, Vol. 1844, pp. 555–569 (2017).

19. Savenko, O., Lysenko, S., Nicheporuk, A., Savenko, B.: Approach for the Unknown Metamorphic Virus Detection. In: 9-th IEEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems. Technology and Applications, pp. 453–458 (2017).

20. Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A. and Nicheporuk, A.: A technique for detection of bots which are using polymorphic code. In: 21st International Conference, CN, Springer, Brunów, Poland, pp. 265-276 (2014)

21. Kondratenko, Y., Kondratenko, N.: Soft Computing Analytic Models for Increasing Efficiency of Fuzzy Information Processing in Decision Support Systems. Chapter in book: Decision Making: Processes, Behavioral Influences and Role in Business Management, R. Hudson (Ed.), Nova Science Publishers, New York, 41-78 (2015)

22. Savenko O.S Research of methods of antiviral diagnostics of computer networks / O.S Savenko, S.M Lysenko // Visnyk of Khmelnytsky National University. Technical sciences. - 2007. - № 2, v. 2. - P. 120–126.(in Ukrainian)

23. Savenko O.S., Payuk V.P., Savenko B.O, Kashtalyan A.S. Models of undocumented software bookmarks in local computer networks / Measuring and computing equipment in technological processes / 2019. - №2. - P.84-90.(in Ukranian)

24. Savenko O.S. Model of the process of searching for Trojan programs in a personal computer / O.S. Savenko, S.M. Lysenko // Radio electronic and computer systems. - 2008. - №7. - P.87-92. (in Ukrainian)

Надійшла / Paper received: 02.09.2020 Надрукована / Paper Printed : 01.12.2020