УДК 004.05:621.31
DOI: 10.31673/2409-7292.2024.020005

Барабаш О. В., Свинчук О. В., Бандурка О. І.

ПРОГРАМНЕ ЗАБЕПЕЧЕННЯ КОНТРОЛЮ СПРАВНОГО СТАНУ ІНФОРМАЦІЙНИХ СИСТЕМ В ЕНЕРГЕТИЧНІЙ ГАЛУЗІ ДЛЯ ЗАБЕЗПЕЧЕННЯ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ

Сьогодні установи енергетичної галузі все більше залежать від функціонування інформаційних систем. Завдяки швидкому розвитку технологій, інформаційні системи стають все більш складними та масштабними. Вони включають в себе велику кількість компонентів, які взаємодіють між собою, і важливо контролювати справний стан і забезпечувати їх надійну роботу та стійкість. Відмова або порушення роботи таких систем може призвести до серйозних фінансових, соціальних та технічних проблем. Враховуючи ці фактори, дослідження процесу моніторингу параметрів функціонування та контролю справного стану інформаційної системи є актуальним завданням. Створено систему моніторингу функціональних параметрів та контролю справного стану модулів інформаційної системи електростанції на основі алгоритму самодіагностування тестовим методом для забезпечення функціональної стійкості. Система виконує постійний контроль елементів системи тестовим методом з блукаючим діагностичним ядром. При виникненні відмови, пошкодження або збою, забезпечується локалізація відмовленого елемента, зміна структури системи шляхом перерозподілу задач на справні вузли, видача інформації про виявлені відмови робочому персоналу системи для ініціювання процесу ремонту та відновлення працездатності відмовленого компонента інформаційної системи. Для підтвердження правильності розробленого програмного продукту проведено математичне моделювання процесу діагностування інформаційної системи для різної кількості вузлів. Для аналізу результатів перевірок заданий рівень достовірності діагностування, який виконує роль ознаки припинення накопичування результатів перевірок в пам'яті модуля, що буде виконувати зазначений алгоритм.

Ключові слова: інформаційна система, база даних, моніторинг, програмне забезпечення, функціональна стійкість, зовнішні та внутрішні дестабілізуючі фактори, діагностування, тестовий контроль, електростанція, ймовірність.

Вступ

Надійна енергетика в країні є головним фактором для нормального функціонування усіх сфер її діяльності. Інформаційні системи електростанцій в сучасних умовах функціонують в умовах впливу зовнішніх та внутрішніх дестабілізуючих факторів. За негативного впливу модулі системи можуть виходити з ладу, але незважаючи на це, системи повинні функціонувати в автономному режимі протягом заданого часу. Таку умову функціонування можна виконати завдяки забезпеченню властивості функціональної стійкості.

Функціональна стійкість – це можливість функціонування інформаційної системи протягом вказаного часу під впливом зовнішніх і внутрішніх дестабілізуючих факторів [1]. Під зовнішніми та внутрішніми дестабілізуючими факторами розуміються відмови, збої модулів системи, механічні пошкодження, помилки обслуговуючого персоналу. Поняття функціональної стійкості було введено професором Машковим О.А. та в подальшому розвинуто Барабашом О.В. для складних технічних систем. Основними етапами забезпечення функціональної стійкості є виявлення модуля або декількох модулів, які відмовили в системі з подальшим їхнім діагностуванням та відновлення функціонування інформаційної системи електростанції.

Отже, процес контролю справного стану та моніторингу параметрів функціонування багатомодульних інформаційних систем в сучасному світі стає досить актуальним завданням.

Аналіз літературних даних та постановка проблеми

Вирішенню проблеми забезпечення функціональної стійкості також присвячено низку наукових праць таких науковців, як Машков О.А., Барабаш О.В., Кравченко Ю.В., Обідін Д.М., Собчук В.В., Мусієнко А.П. та інші. Розв'язання задач функціональної стійкості багатомодульних інформаційних систем відповідального призначення критичної інфраструктури значною мірою залежить від можливості моніторингу та прогнозування їх технічного стану [2], визначення основних вимог та критеріїв для функціонально стійких складних систем [3]. У роботах [4] запропоновані методи для забезпечення відмовостійкості

інформаційної системи та метод для діагностики її несправностей [5]. На основі отриманих результатів запропоновано нові моделі діагностики інформаційної системи для виявлення несправних процесорів [6].

В [7] описано методологію побудови ефективної системи самодіагностики інформаційних систем на прикладі підприємств металургійної та енергетичної промисловості. Наведено методику організації та здійснення самодіагностики, механізми виявлення, а також ідентифікацію та локалізацію несправних модулів. Проведено аналіз застосування ієрархічного підходу для організації засобів забезпечення функціональної стійкості [8].

Для об’єктів критичної інфраструктури запропоновано метод побудови закону управління безпекою критичних об'єктів інфраструктури в умовах зовнішніх неконтрольованих впливів [9]. Описується стійкість безпеки інформаційних систем, де визначається стійкість безпеки інформаційних систем як динамічна здатність системи реагувати на атаку та відновлюватися після неї [10]. З метою автоматизації системи контролю та діагностування мікропроцесорних систем запропоновано реалізувати принцип самодіагностування, в основу якого покладено ідеї штучного інтелекту [11].

Проводилися дослідження функціональної стійкості структури інформаційної телекомунікаційної мережі [12] та навігаційного забезпечення повітряних суден цивільної авіації з визначенням нових підходів [13]. Науковцями розглядаються різноманітні архітектури інформаційних систем з точки зору функціональної стійкості, а також вплив її на мережеві ресурси та мережеві сервіси [14].

Отже, для всіх складних технічних систем дуже важливо здійснювати моніторинг параметрів функціонування та справного стану, оскільки прогнозування дозволяє розв’язувати задачу визначення оптимальних моментів контролю, у проміжках між якими забезпечуватиметься властивість функціональної стійкості системи.

Метою дослідження є створення системи моніторингу функціональних параметрів та контролю справного стану обчислювальних вузлів інформаційної системи електростанції на основі тестового діагностування для забезпечення функціональної стійкості.

Основна частина

Функціональна стійкість будь-якої інформаційної системи забезпечується комплексом процесів та механізмів, які здатні підтримувати нормальну роботу системи навіть в умовах відмов або негативних впливів. Тому необхідно постійно проводити моніторинг її справного стану, який ідентифікує, аналізує та вчасно реагує на будь-які проблеми, забезпечуючи надійну та безперебійну роботу. Для досягнення функціональної стійкості системи важливо вибрати відповідний алгоритм діагностування системи, який врахує всі особливості системи та дозволить визначити, чи працюють модулі системи належним чином, оперативно виявити несправності та вжити відповідні заходи для забезпечення надійності та продуктивності системи.

Серед існуючих методів діагностування обрано метод тестового діагностування, який за принципом побудови поділяється на два типи:

тестове діагностування з централізованим діагностичним ядром використовує централізований контроль, тобто діагностичне ядро знаходиться в центральному вузлі системи, що дозволяє ефективно керувати та контролювати процес діагностики;

тестове діагностування з блукаючим діагностичним ядром розподіляє завдання діагностування між вузлами, де діагностичні функції розподілені між різними вузлами системи, що сприяє розділенню навантаження та підвищенню продуктивності.

Обидва методи мають свої переваги та недоліки і вибір залежить від конкретних вимог та характеристик системи. Проведено порівняння цих методів і вирішено було обрати тестове діагностування за принципом блукаючого діагностичного ядра. За даним методом діагностування складається з елементарних перевірок вузлами інших вузлів системи, які виконуються у випадкові моменти часу. Кожен вузол локально в своїй пам’яті зберігає

матрицю з результатами елементарних тестів. Обмін діагностичною інформацією про результати перевірок проводиться між вузлами на основі способу умовної передачі результатів елементарних перевірок. Кожен вузол, отримуючи діагностичну інформацію, формує ознаку достатності щодо алгоритму дешифрації отриманої діагностичної інформації. Як ознака достатності використовується перевірка умови досягнення ліміту кількості елементарних тестів та перевірка чи всі вузли було протестовано. При задоволенні зазначеної ознаки достатності, вузол, на якому позитивно виконана ознака, виконує алгоритм дешифрації діагностичної інформації та визначає технічний стан усіх вузлів розподіленої інформаційної системи. Процес визначення технічного стану вузла залежить від умовної перевірки отриманого результату з достовірністю діагностування системи. Достовірність діагностування задається користувачем або використовується значення за замовчуванням [2].

Схематичне зображення архітектури системи зображено у рисунку 1.

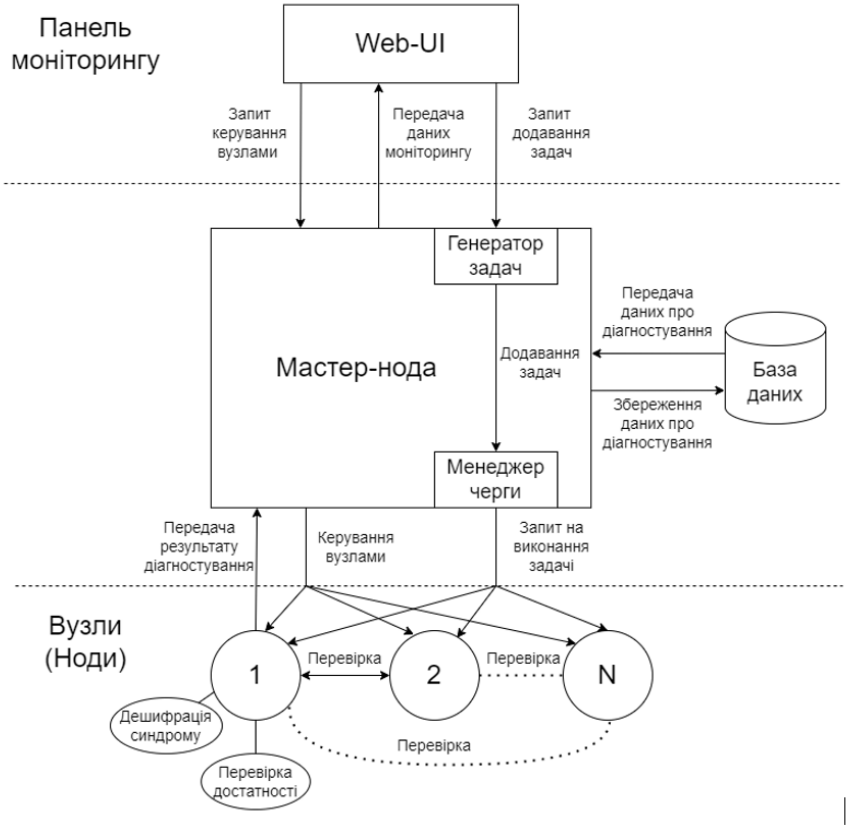

Рис. 1. Архітектура системи

Система поділяється на чотири блоки. Зв'язок між ними забезпечується за допомогою HTTP-запитів та web-сокетів: панель моніторингу, мастер-нода, база даних та мережа вузлів.

Панель моніторингу (Web-GUI) є односторінковим веб-застосунком та забезпечує демонстрацію даних моніторингу користувачу та передачу запитів користувача до наступного блоку системи. В якості бази даних в системі використовується PostgreSQL. Вона зберігає інформацію про стан вузлів системи, про події, які відбулись в системі, та результати діагностування системи.

Мастер-нода – це головний вузол в системі моніторингу, який забезпечує взаємодію користувача із мережею вузлів, з базою даних системи, а також зв’язок веб-інтерфейсу із системою. До будови мастер-ноди входить модуль BullMQ, який відповідає за генерацію та розподілену обробку задач. BullMQ має чергу задач і розподіляє задачі з черги на вільні вузли в мережі. Модуль було налаштовано таким чином, що кожен вузол може обробляти до трьох

задач одночасно, оптимізуючи використання ресурсів. Якщо всі вузли зайняті повністю, BullMQ автоматично перенаправляє нові задачі до черги, де вони залишаються у черзі до появи вільного вузла. Цей модуль забезпечує ефективне використання обчислювальних потужностей системи та уникнення перевантаження окремих вузлів.

Мережа вузлів складається із елементарних вузлів, які забезпечують виконання задач в системі, а також виконують алгоритм самодіагностування на справність самих себе. Вузли з'єднані між собою у віртуальній мережі та взаємодіють за допомогою HTTP-запитів. Кожен вузол може виконувати обчислення та діагностування, забезпечуючи розподілені обчислення та взаємодію в мережі.

Взаємодія між мастер-нодою та панеллю моніторингу здійснюється через стандартну архітектуру REST API. Отримання даних про компоненти системи відбувається за допомогою HTTP-запитів, в яких кожен ресурс ідентифікується унікальним (Uniform Resource Identifier). Цей підхід сприяє прозорій та ефективній взаємодії між частинами системи, а також надає стандартизований інтерфейс для обміну даними.

Підхід REST API, використаний у системі, робить її простішою, легшою для розробки та масштабованою. Використання HTTP-запитів і стандартних методів HTTP (GET, POST, PATCH, DELETE) робить комунікацію зрозумілою та простою. REST API дозволяє легко інтегрувати систему з іншими додатками та сервісами. REST API дозволяє передавати дані у різних форматах, таких як JSON або XML, що забезпечує інформаційну гнучкість і можливість використовувати дані на різних платформах. Спрощений протокол комунікації у вигляді HTTP-запитів полегшує валідацію та тестування функціоналу системи.

На рисунку 2 представлена діаграма прецендентів взаємодії користувача із системою.

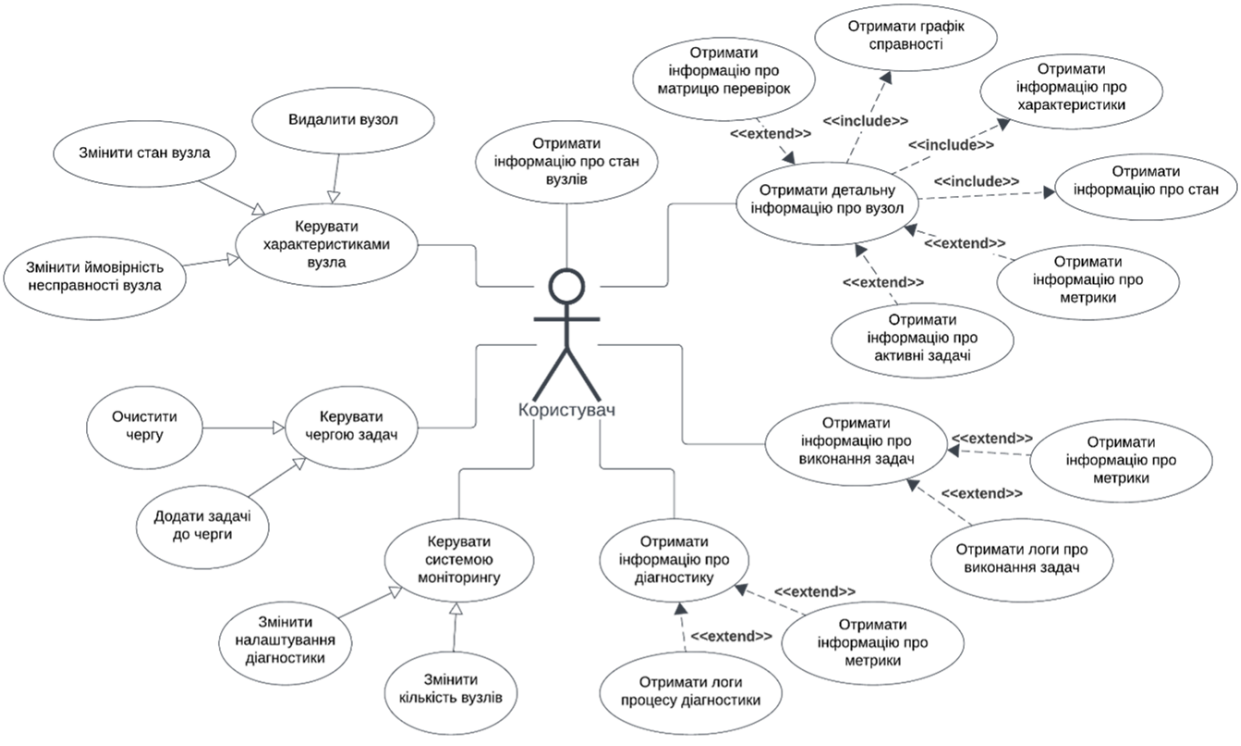

Рис. 2. Діаграма прецендентів взаємодії користувача із системою

З урахуванням труднощів у розгортанні розробленої системи в реальних умовах, вирішено використовувати віртуальне розгортання системи за допомогою платформи упакування програмного забезпечення Docker. Кожен компонент системи реалізований у вигляді ізольованого контейнера, який виконується локально на одному комп'ютері.

Контейнери розміщені в одній спільній мережі і кожен має свою унікальну IP-адресу, що дозволяє кожному елементу системи бачити один одного та взаємодіяти в межах цієї мережі. Це забезпечує ефективну комунікацію між компонентами системи.

Розроблена система дозвляє відстежувати стан різних компонентів інформаційної системи та своєчасно виявляти можливі проблеми або відмови з метою підтримки безперервної роботи системи за допомогою діагностування на основі методів тестового контролю.

На етапі експлуатації інформаційної системи здійснюється функціонування в штатному режимі, при якому вона виконує призначені їй функції – виконання множини задач. Система, частина якої вбудована у вузли інформаційної системи, виконує постійний контроль елементів системи тестовим методом. При виникненні відмови, пошкодження або збою, забезпечується локалізація відмовленого елемента, зміна структури системи шляхом перерозподілу задач на справні вузли, видача інформації про виявлені відмови робочому персоналу системи для ініціювання процесу ремонту та відновлення працездатності відмовленого компонента розподіленої інформаційної системи. Головною сторінкою програми є Панель Моніторингу, на якій розміщена загальна інформація про стан системи (рис. 3).

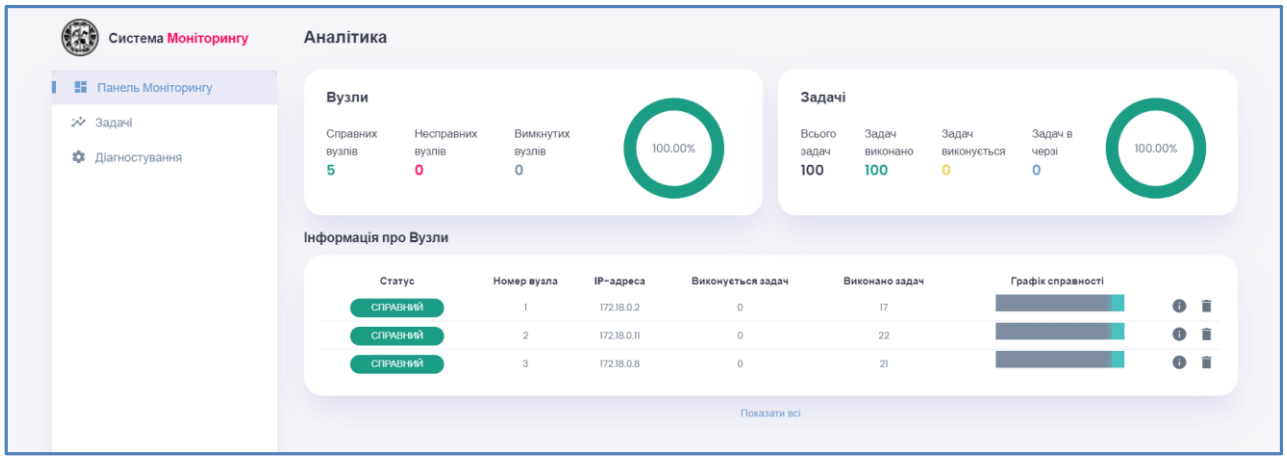

Рис. 3. Меню «Панель моніторингу»

Панель місить три основних блоки:

вузли – у блоці наведено інформацію про кількість справних, несправних та вимкнутих вузлів у системі; справа розміщена кругова інформація, яка вказує відсоток справних вузлів;

задачі – у блоці наведено чотири текстових поля, які інформують про кількість задача загалом, виконаних задач, задач, які виконуються, та задач в черзі; справа розміщена кругова інформація, яка вказує відсоток виконаних задач;

таблиця з інформацію про вузли інформує про справність чи несправність конкретного вузла, а також додатково має такі колонки: номер вузла (id), IP-адреса вузла в комп'ютерній мережі, кількість задач, які виконуються прямо зараз, загальна кількість виконаних задач цим вузлом, графік стану вузла за останні 30 хвилин.

В меню «Задачі» зосереджений весь функціонал, що стосується логіки виконання задач в системі (рис. 4).

В даному меню є блок про виконання задач у системі та блок, що відображає більш детальну інформацію про виконання, а саме: про прогрес виконання задач, завантаженість процесора сервера, середня завантаженість за 1, 5 та 15 хвилин (Load Average) та середній час виконання задачі в цілому усіх вузлів разом. Середня завантаженість показує потребу у виконуваних потоках у вигляді усередненої кількості виконуваних та очікуваних потоків:

якщо значення дорівнюють 0.0, то система в стані простою;

якщо середнє значення для 1 хвилини вище, ніж для 5 або 15, навантаження зростає;

якщо середнє значення для 1 хвилини нижче, ніж для 5 або 15, навантаження знижується;

якщо значення навантаження вищі за кількість процесорів, то у вас можуть бути проблеми з продуктивністю (залежно від ситуації).

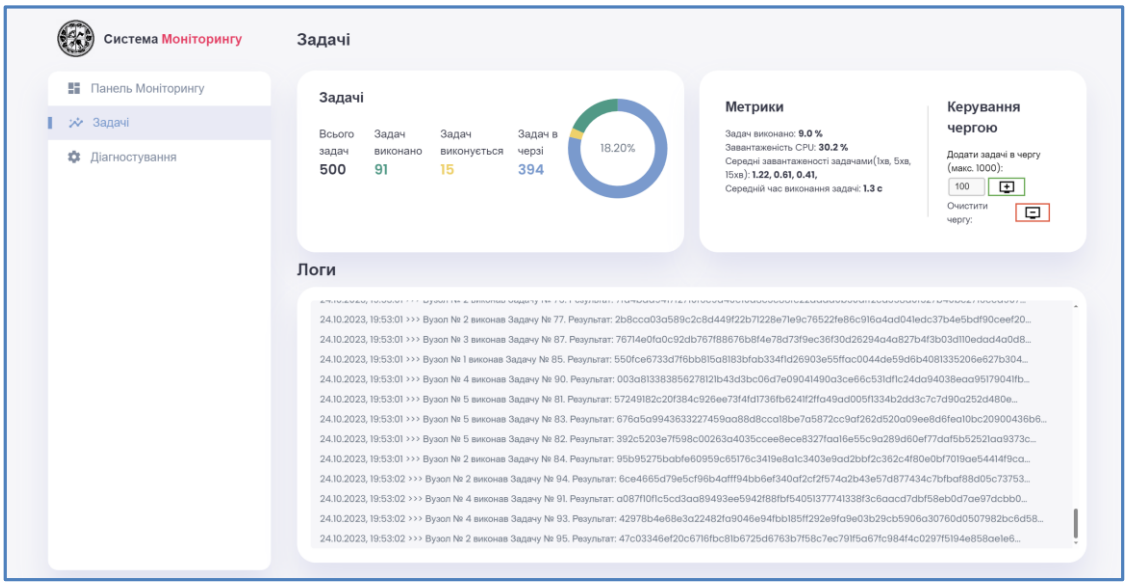

Рис. 4. Меню «Задачі»

У меню «Діагностування» відображається інформація про алгоритм діагностування вузлів системи (рис. 5). На цій вкладці дублюється блок про справні вузли у системі, а також розміщено новий блок, що відображає більш детальну інформацію про діагностування. У блоці про справні вузли з'явилось дві кнопки: «Показати результати останнього діагностування» і «Налаштування діагностики».

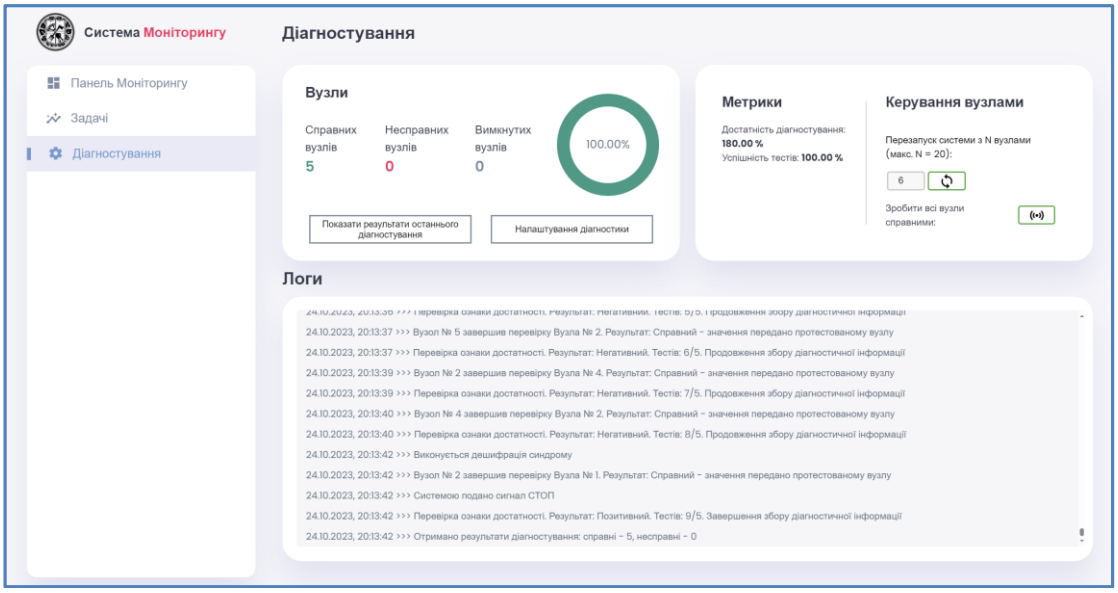

Рис. 5. Меню «Діагностування»

При натисканні на кнопку «Показати результати останнього діагностування» відкривається вікно з інформацією про останній результат алгоритму діагностування – час, коли такий результат було отримано, матриця результатів перевірок та ймовірності справності кожного вузла. В частині блоку «Керування вузлами» можна здійснити перезапуск системи з визначеною кількістю вузлів у мережі. Нижче під блоками розміщені логи – журнал подій за

останні 5 хвилин, що стосуються алгоритму діагностування. В журналі зберігається час події, і що за подія відбулася:

вузол виконав елементарну перевірку;
перевірка ознаки достатності;
результат діагностування;
виконання дешифрації синдрому;
подано сигнал «старт»;
подано сигнал «стоп».

Було промодельовано розроблений програмний продукт для кількості вузлів $N = 15$, $N = 20$, $N = 25$ та $N = 30$. Передбачається, що всі вузли, що входять до системи, зв'язані з іншими вузлами системи, тобто існує принаймні один маршрут передачі інформації між будь-якою парою вузлів системи. Також всі вузли системи не є абсолютно надійними, тому приймається, що на момент початку процедури діагностування, кожний вузол має певну апріорну ймовірність працездатного стану, яка позначена p та варіюється під час моделювання в таких значеннях: 0,8; 0,85; 0,9; 0,95. Для аналізу результатів перевірок задається заданий рівень достовірності діагностування $D_{зад}$, який виконує роль ознаки припинення накопичування результатів перевірок в пам'яті модуля, що буде виконувати зазначений алгоритм. Задана достовірність варіюється під час моделювання із такими значеннями: 0,8; 0,85; 0,9; 0,95; 0,98.

За результатами моделювання визначено зміну достовірності діагностування від числа вузлів, що відмовили за різних значень, заданого рівня достовірності $D_{зад}$ та p. Аналіз графіків показує, що за умови відсутності модулів, що відмовили $N_{від}$=0, в усіх випадках буде 100-відсоткова достовірність D=1:

при $N_{від}$<5 достовірність діагностування наближається до 1: D≥0,98 . За умови значної кількості відмов $N_{від}$ = 6 … 12 для системи із N=15 модулів, достовірність діагностування в будь-яких умовах вище ніж $D_{зад}$;

при $N_{від}$<7 достовірність діагностування наближається до 1: D≥0,98 . За умови значної кількості відмов $N_{від}$ = 8 … 16 для системи із N=20 модулів, достовірність діагностування в будь-яких умовах вище ніж $D_{зад}$;

при $N_{від}$<9 достовірність діагностування наближається до 1: D≥0,98 . За умови значної кількості відмов $N_{від}$ = 10 … 20 для системи із N=25 модулів, достовірність діагностування в будь-яких умовах вище ніж $D_{зад}$;

при $N_{від}$<11 достовірність діагностування наближається до 1: D≥0,95 . За умови значної кількості відмов $N_{від}$ = 14 … 22 для системи із N=30 модулів, достовірність діагностування в будь-яких умовах вище ніж $D_{зад}$ (рис. 6).

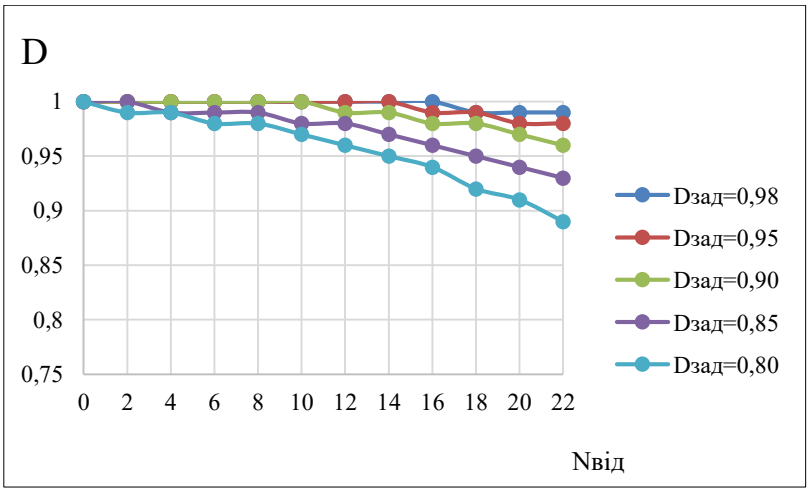

Рис. 6. Залежність достовірності діагностування D від числа модулів, що відмовили $N_{від}$, p=0,95, N =30

Результати моделювання залежності достовірності діагностування D від кількості відмов у системі $N_{від}$ за умови варіювання розмірності системи N та апріорної імовірності працездатного стану вузлів p та заданої достовірності $D_{зад}$ дозволяє зробити висновки:

достовірність діагностування практично не змінюється внаслідок збільшення числа вузлів системи N;

значення достовірності суттєво залежать від числа відмов $N_{від}$ – при збільшенні числа $N_{від}$ зменшується достовірність діагностування D;

залежність достовірності від апріорної імовірності працездатного стану вузлів p показує незначне підвищення достовірності із зростанням значення p.

Отже, дані результати підтверджують основну властивість тестового діагностування – можливість діагностування із достовірністю не нижче заданої.

Висновки

Створено систему моніторингу функціональних параметрів та контролю справного стану обчислювальних пристроїв інформаційної системи на основі алгоритму самодіагностування тестовим методом з блукаючим діагностичним ядром для забезпечення функціональної стійкості. Представлена архітектура системи з детальним описок кожного блоку, зв'язок між якими забезпечується за допомогою HTTP-запитів та web-сокетів: панель моніторингу, мастер-нода, база даних та мережа вузлів. Взаємодія між мастер-нодою та панеллю моніторингу здійснюється через стандартну архітектуру REST API. Отримання даних про компоненти системи відбувається за допомогою HTTP-запитів, в яких кожен ресурс ідентифікується унікальним (Uniform Resource Identifier). Цей підхід сприяє прозорій та ефективній взаємодії між частинами системи, а також надає стандартизований інтерфейс для обміну даними. Розроблена система моніорингу дозвляє відстежувати стан різних компонентів інформаційної системи та своєчасно виявляти можливі проблеми або відмови з метою підтримки безперервної роботи системи за допомогою діагностування на основі методів тестового контролю.

Для підтвердження правильності розробленого програмного продукту проведено математичне моделювання процесу діагностування інформаційної системи для різної кількості вузлів. Для аналізу результатів перевірок задається заданий рівень достовірності діагностування $D_{зад}$, який виконує роль ознаки припинення накопичування результатів перевірок в пам’яті модуля, що буде виконувати зазначений алгоритм. За результатами моделювання визначено, що достовірність діагностування практично не змінюється внаслідок збільшення числа вузлів системи N; значення достовірності суттєво залежать від числа відмов $N_{від}$ – при збільшенні числа $N_{від}$ зменшується достовірність діагностування D;
залежність достовірності від апріорної імовірності працездатного стану вузлів p показує незначне підвищення достовірності із зростанням значення p.

Перспективи подальших досліджень вбачаються у покращенні системи завдяки додаванню нового функціоналу, а саме впровадження штучного інтелекту для забезпечення функціональної стійкості інформаційної системи електростанції.

Перелік посилань

1. Собчук, В. В., Барабаш, О. В., Мусієнко, А. П. Основи забезпечення функціональної стійкості інформаційних систем підприємств в умовах впливу дестабілізуючих факторів: монографія. Київ: Міленіум, 2022. 272 с. ISBN: 973-966-8063-82-3 https://www.researchgate.net/publication/363474851_Basis_for_functional_stability_of_information_systems_businesses_under_the_influence_of_destabilizing_factors

2. Barabash, O., Sobchuk, V., Musienko, A., Laptiev, O., Bohomia, V., Kopytko, S. System Analysis and Method of Ensuring Functional Sustainability of the Information System of a Critical Infrastructure Object. In: Zgurovsky, M., Pankratova, N. (eds) System Analysis and Artificial Intelligence. Studies in Computational Intelligence, 2023. Vol 1107. Springer, Cham. P. 117-192. https://doi.org/10.1007/978-3-031-37450-0_11

3. Барабаш, О. В. Побудова функціонально стійких розподілених інформаційних систем. Київ: НАОУ, 2004. 226 с. https://bit.ly/3wM5tDL

4. Peng, S.-L., Lin, C.-K., Tan, J. J. M., and Hsu, L.-H. The g-Good-Neighbor Conditional Diagnosability of Hypercube under PMC Model. Applied Mathematics and Computation, 2012. Vol. 218, no. 21. P. 10406-10412. https://doi.org/10.1016/j.amc.2012.03.092

5. Yuan, J., Liu, A., Ma, X., Liu, X., Qin, X., and Zhang, J. The g-Good-Neighbor Conditional Diagnosability of k-Ary n-Cubes under the PMC Model and MM Model. IEEE Transactions on Parallel and Distributed Systems, 2015. Vol. 2, no. P. 1165-1177. http://doi.org/10.1109/TPDS.2014.2318305

6. Ren, Y., Wang, S. Some Properties of the g-Good-Neighbor (g-Extra) Diagnosability of a Multiprocessor System. American Journal of Computational Mathematics. 2016. Vol. 6, no. 3. P. 259-266. http://doi.org/10.4236/ajcm.2016.63027

7. Собчук, А. В., Олімпієва, Ю. І. Застосування нейромереж для забезпечення функціональної стійкості виробничих процесів. Телекомунікаційні та інформаційні технології. К.: ДУТ, 2020. № 2 (67). С. 13-28. http://doi.org/10.31673/2412-4338.2020.021328

8. Sobchuk, V., Barabash, O., Musienko, A., Svynchuk, O. Adaptive accumulation and diagnostic information systems of enterprises in energy and industry sectors. E3S Web of Conferences. 2021. Vol. 250. P. 82-87. https://doi.org/10.1051/e3sconf/202125008002

9. Laptiev, O., Barabash, O., Tsyganivska, I., Obidin, D., Sobchuk, A. The Method of Construction of the Law of Safety Management of Critical Infrastructure Objects Under the Conditions of External Uncontrolled Influences. CEUR Workshop Proceedings. 2023. Vol. 3624. P. 291–300. https://ceur-ws.org/Vol-3624/Paper_24.pdf

10. Goel L., Russell D., Williamson S. and Zhang, J.Z. Information systems security resilience as a dynamic capability. Journal of Enterprise Information Management.2023. Vol. 36, no. 4. P. 906-924. https://doi.org/10.1108/JEIM-07-2022-0228

11. Тюлюпа, С. В., Самохвалов, Ю. Я., Хусаінов, П. В., Штатенко, С. С. Самодіагностування як спосіб підвищення кіберстійкості термінальних компонентів технологічної системи. Кібербезпека: освіта, наука і техніка. № 2 (22). 2023. С. 134-147. https://doi.org/10.28925/2663-4023.2023.22.134147

12. Обідін, Д. М. Оцінка функціональної стійкості інформаційно-телекомунікаційних мереж на основі автоматизованих систем управління. Наука і техніка Повітряних Сил Збройних Сил України. 2014. № 1. С. 167-169. http://nbuv.gov.ua/UJRN/Nitps_2014_1_40

13. Калашник, Г. А., Обідін, Д. М., Калашник, М. А. Забезпечення стійкого функціонування засобів навігації літальних апаратів під впливом зовнішніх дестабілізуючих факторів. Системи обробки інформації. 2016. № 3 (140). С. 52-56. http://doi.org/10.30748/nitps.2021.44.07

14. Dovgiy, S., Kopiika, O., Kozlov, O. Architectures for the Information Systems, Network Resources and Network Services. In: CEUR Workshop Proceedings: CPITS-II-1: Cybersecurity Providing in Information and Telecommunication Systems II, 2021. Vol. 3187. P. 293-301. https://ceur-ws.org/Vol-3187/

Надійшла 14.04.2024