

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

“___” червня 2026 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою

«Системне програмування та спеціалізовані комп'ютерні системи»

спеціальності 123 «Комп'ютерна інженерія»

**на тему: «Програма побудови та перевірки GL-моделей складних небазових відмовостійких
багатопроцесорних систем»**

Виконав (-ла):

студент (-ка) IV курсу, групи КВ-22
(шифр групи)

Кошулько Владислав Віталійович

(прізвище, ім'я, по батькові)

(підпис)

Керівник доцент каф. СПіСКС, к.т.н., Морозов К.В.

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант з нормоконтролю, доц. каф. СПіСКС, к.т.н., доц. Клятченко Я.М.

(назва розділу)

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

(підпис)

Рецензент д.т.н., проф., проф. каф. ОТ ФІОТ Сергієнко А.М.

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій дипломній роботі немає
запозичень з праць інших авторів без
відповідних посилань.

Студент

(підпис)

Київ – 2026 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРОГРАМНИХ СИСТЕМ ТА ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Системне програмування та спеціалізовані комп'ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

«___» _____ 202_р.

**ЗАВДАННЯ
на дипломний проєкт студента**

Кошулько Владислав Віталійович
(прізвище, ім'я, по батькові)

1. Тема проєкту «Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем»

керівник проєкту доцент каф. СПіСКС, к.т.н., Морозов К.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету № 1984-С від «28» травня 2026 р.

2. Термін подання студентом проєкту _____

3. Вихідні дані до проєкту: Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем

4. Зміст пояснювальної записки:

- 1) Аналіз існуючих рішень та актуальність теми дипломного проєкту;
- 2) GL-МОДЕЛЬ;
- 3) РОБОТА ПРОГРАМИ.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо):

- 1) Побудова базової моделі. Схема алгоритму.
- 2) Перевірка моделі на правильність. Схема алгоритму.
- 3) Побудова небазової моделі. Схема алгоритму.
- 4) Крок формування реберної функції базової моделі. Схема алгоритму.
- 5) Слайди презентації.

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Клятченко Я.М., доц. каф. СПіСКС, К.Т.Н		

7. Дата видачі завдання ____ 202_ р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Видача завдання на дипломне проєктування	08.01.2026	
2.	Вивчення літератури за тематикою проєкту	17.04.2026	
3.	Розроблення та узгодження технічного завдання	28.04.2026	
4.	Аналіз існуючих рішень	06.05.2026	
5.	Підготовка матеріалів першого розділу дипломного проєкту	11.05.2026	
6.	Підготовка матеріалів другого та третього розділів дипломного проєкту	14.05.2026	
7.	Підготовка графічної частини дипломного проєкту	20.05.2026	
8.	Оформлення документації дипломного проєкту	22.05.2026	
9.	Попередній огляд матеріалів проєкту на кафедрі	29.05.2026	

Студент

(підпис)

Владислав КОШУЛЬКО

(ім'я, ПРІЗВИЩЕ)

Керівник проєкту

(підпис)

Костянтин МОРОЗОВ

(ім'я, ПРІЗВИЩЕ)

АНОТАЦІЯ

Кваліфікаційна робота включає пояснювальну записку (54с., 5рис., 1 табл., 2 додатки).

Об'єкт розробки – Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем.

Розроблена програма дозволяє будувати коректні GL-моделі складних небазових багатопроцесорних систем, перевіряє створені моделі на коректність В процесі розробки:

- Розглянуто, що собою представляє відмовостійка багатопроцесорна система та сфери використання
- Проаналізовано існуючі методи моделювання ВБС
- Підтверджено роботу алгоритмів для побудови базових та небазових GL-моделей
- Експериментально перевірено побудовані моделі на коректність

Ключові слова: GL-МОДЕЛЬ, ВБС, МВР, ГРАФ, ГРАФО-ЛОГІЧНА МОДЕЛЬ, ВІДМОВОСТІЙКІСТЬ, ПЕРЕВІРКА НАДІЙНОСТІ

ABSTRACT

The qualification work includes an explanatory report (54 pages, 5 figures, 1 tables, 2 appendices).

Object of development – A software program for constructing and verifying GL models of complex non-basic fault-tolerant multiprocessor systems.

The developed program enables the construction of correct GL models of complex non-basic multiprocessor systems and performs validation of the created models for correctness.

During the development process:

- The concept of fault-tolerant multiprocessor systems and their areas of application were examined
- Existing methods for modeling fault-tolerant multiprocessor systems (FTMS) were analyzed
- The functionality of algorithms for constructing both basic and non-basic GL models was confirmed
- The constructed models were experimentally verified for correctness

Keywords: GL model, FTMS (fault-tolerant multiprocessor systems), MLE (Minimum Lost Edges), graph, graph-logical model, fault tolerance, reliability verification

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
	A4	ІАЛЦ.467200.002 ТЗ	Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем.	4		
			Технічне завдання			
	A4	ІАЛЦ. 467200.003 ТП	Програма побудови та перевірки GL-моделей відмовостійких багатопроцесорних систем.	2		
			Відомість технічного проєкту			
	A4	ІАЛЦ. 467200.004 ПЗ	Програма побудови та перевірки GL-моделей відмовостійких багатопроцесорних систем.	54		
			Пояснювальна записка			

Поз.	Формат	ПОЗНАЧЕННЯ	НАЙМЕНУВАННЯ	Кількість аркушів	№ прим.	Примітки
A1	IАЛЦ.467200.005 Д1	Побудова базової моделі	Схема алгоритму	1		
A1	IАЛЦ.467200.006 Д2	Перевірка моделі на правильність	Схема алгоритму	1		
A1	IАЛЦ.467200.007 Д3	Побудова небазової моделі	Схема алгоритму	1		
A1	IАЛЦ.467200.008 Д4	Крок формування реберної Функції базової моделі Схема алгоритму	Текст пояснювальної записки. Графічний матеріал	1		
Disk CD-ROM						
Zмін.	Арк.	№ докум.	Підпис	Дата		
IАЛЦ.467200.001 ОА					Арк.	2

ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ	2
2. ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4. ДЖЕРЕЛА РОЗРОБКИ.....	2
5. ТЕХНІЧНІ ВИМОГИ.....	3
5.1 Вимоги до програмного продукту, що розробляється	3
5.2 Вимоги до апаратного забезпечення	3
5.3 Вимоги до програмного та апаратного забезпечення користувача	3
6. ЕТАПИ РОЗРОБКИ	4

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем».

Галузь застосування: відмовостійкі багатопроцесорні системи керування складними об'єктами.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даної роботи є створення програми побудови та перевірки GL-моделей.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література з побудови алгоритмів моделювання складних небазових систем та їх різновиди

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється:

- Інтерфейс для користувача
- Можливість перевірки моделі
- перевірка роботи програми
- Можливість задавати параметри

5.2 Вимоги до апаратного забезпечення:

- Процесор з 4 ядрами або більше
- 8 Гб оперативної пам'яті або більше
- Мова програмування C++ та Python

5.3 Вимоги до програмного та апаратного забезпечення користувача:

- Операційна система Windows. Linux та macOS

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів
1.	Вивчення літератури за тематикою проєкту	17.04.2026
2.	Розроблення та узгодження технічного завдання	28.04.2026
3.	Аналіз існуючих рішень	06.05.2026
4.	Підготовка матеріалів першого розділу дипломного проєкту	11.05.2026
5.	Підготовка матеріалів другого розділу дипломного проєкту	14.05.2026
6.	Підготовка графічної частини дипломного проєкту	20.05.2026
7.	Оформлення документації дипломного проєкту	22.05.2026
8.	Попередній огляд матеріалів диплому на кафедрі	29.05.2026

[illegible]

[illegible]

Пояснювальна записка
до дипломного проєкту

на тему: Програма побудови та перевірки GL-моделей складних небазових
відмовостійких багатопроцесорних систем

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ.....	3
ВСТУП	4
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА АКТУАЛЬНІСТЬ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ	7
1.1 Складні небазові відмовостійкі багатопроцесорні системи	7
1.2 Існуючі методи та їх недоліки	12
1.3 Комбіновані статистичні підходи	17
1.4 Висновки до розділу	19
2 GL-МОДЕЛЬ	21
2.1 Графо-логічна модель	21
2.2 Базова GL-модель	27
2.3 Побудова базової моделі	28
2.4 Небазова модель	29
2.5 Побудова небазової моделі	31
2.6 Модифікація небазової моделі	34
2.6 Перевірка коректності моделі	36
3 РОБОТА ПРОГРАМИ	42
3.1 Розробка програми	42
3.2 Методи та принципи, використані під час розробки	45
3.3 Особливості програми	47
3.4 Аналіз отриманих результатів	48
ВИСНОВОК	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	53
ДОДАТКИ:	

ДОДАТОК 1. Копії графічних матеріалів

ІАЛЦ. 467200.005 Д1 Побудова базової моделі.

Схема алгоритму.

ІАЛЦ. 467200.006 Д2 Перевірка моделі на правильність.

Схема алгоритму.

ІАЛЦ. 467200.007 Д3 Побудова базової моделі.

Схема алгоритму.

ІАЛЦ. 467200.008 Д4 Крок формкування реберної функції базової моделі. Схема алгоритму.

ДОДАТОК 2. Слайди презентації

					ІАЛЦ. 467200.004 ПЗ			
Змін	Арк.	№ докум.	Підпис	Дата	<i>Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем</i> Пояснювальна записка	Літ.	Аркуш	Аркушів
Розробив		Кошулько В.В.					2	54
Перевірив		Морозов К.В.						
Н. контроль		Клятченко Я.М.				КПІ ім. Ігоря Сікорського, ФПСІМ КВ-22		
Затвердив		Романкевич В.О.						

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

GL - модель це графо-логічна модель багатопроцесорної системи

ВБС - Відмовостійка багатопроцесорна система

МВР - Мінімум втрачених ребер

ООП - об'єктно-орієнтоване програмування

$K(m, n)$ - позначення базової ВБС (модель відмовостійкості)

$K'(m, n)$ - модифікована (небазова) модель

$\varphi(n, m)$ - кількість ребер у базовій GL-моделі

$\psi(m, l)$ - функція втрати ребер

					ІАЛЦ.467200.004 ПЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасний розвиток інформаційних технологій нерозривно пов'язаний із використанням високопродуктивних обчислювальних систем, здатних виконувати великі обсяги обчислень за мінімальний час. Одним із найефективніших напрямів розвитку таких систем є застосування багатопроцесорних архітектур, які забезпечують паралельне виконання завдань та значне підвищення продуктивності. Сьогодні багатопроцесорні комплекси широко використовуються в наукових дослідженнях, промисловій автоматизації, системах керування транспортом, телекомунікаціях, медицині, аерокосмічній галузі та багатьох інших сферах, де важливими є швидкодія, масштабованість і безперервність функціонування.

Зі збільшенням складності архітектури обчислювальних систем зростає й ймовірність виникнення несправностей окремих компонентів. В умовах, коли сучасні інформаційні комплекси можуть містити сотні або навіть тисячі взаємопов'язаних елементів, повністю виключити можливість відмов практично неможливо. Саме тому поряд із продуктивністю особливого значення набуває здатність системи продовжувати виконання своїх функцій навіть за наявності несправних вузлів.

Відмовостійкість є однією з основних характеристик надійної багатопроцесорної системи. Вона визначає можливість збереження працездатності при виникненні відмов процесорів, каналів зв'язку або інших функціональних елементів. Для забезпечення такого режиму роботи застосовуються різні підходи, серед яких резервування ресурсів, дублювання критичних модулів, динамічний розподіл навантаження та автоматична реконфігурація структури системи. Завдяки цьому навіть у разі виникнення несправностей система може продовжувати виконання поставлених завдань із допустимим зниженням продуктивності.

Особливо важливими відмовостійкі системи є для об'єктів підвищеної відповідальності, де наслідки відмов можуть бути критичними. До таких об'єктів

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		4

належать системи управління атомними електростанціями, авіаційною та космічною технікою, медичним обладнанням, транспортною інфраструктурою, а також складними виробничими процесами. Помилки в роботі подібних комплексів можуть призвести не лише до фінансових збитків, а й до виникнення аварійних ситуацій, екологічних катастроф або загрози життю людей.

Дослідження надійності складних багатопроцесорних систем потребує застосування спеціальних математичних методів. Проведення експериментів на реальних об'єктах часто є економічно не вигідним або технічно неможливим, тому значна роль відводиться моделюванню. Використання математичних моделей дозволяє аналізувати поведінку системи в різних умовах експлуатації, визначати критичні елементи структури, оцінювати вплив відмов на працездатність та прогнозувати рівень надійності ще на стадії проектування.

Складність задачі значно зростає під час аналізу небазових відмовостійких багатопроцесорних систем. Для таких систем працездатність визначається не лише кількістю несправних компонентів, а й особливостями їх розташування в структурі, типом взаємозв'язків між елементами, функціональним призначенням окремих вузлів та іншими факторами. У результаті процес побудови моделей стає складним і потребує значних обчислювальних ресурсів.

Одним із перспективних інструментів дослідження подібних систем є графо-логічні моделі. Вони поєднують переваги графового представлення структури системи та логічного опису умов її працездатності. Такий підхід дає можливість формалізувати складні залежності між елементами, виконувати аналіз великої кількості сценаріїв відмов та визначати граничні умови функціонування системи. Завдяки використанню булевих функцій основна складність переноситься на логічний опис, що дозволяє зберігати відносно просту структуру графа навіть для систем великої розмірності.

У зв'язку зі зростанням складності сучасних багатопроцесорних комплексів актуальним стає створення програмних засобів для автоматизованої побудови та перевірки графо-логічних моделей. Автоматизація цього процесу

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		5

дозволяє значно прискорити аналіз відмовостійкості, зменшити ймовірність помилок під час створення моделей, підвищити достовірність отриманих результатів і забезпечити можливість дослідження систем, що складаються з великої кількості взаємопов'язаних компонентів. Саме тому розробка програмних інструментів для побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем є актуальним та важливим напрямом сучасних досліджень у галузі комп'ютерної інженерії.

					ІАЛЦ.467200.004 ПЗ	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА АКТУАЛЬНІСТЬ ТЕМИ ДИПЛОМНОГО ПРОЄКТУ

1.1 Складні небазові відмовостійкі багатопроцесорні системи

У сучасних інформаційно-обчислювальних комплексах дедалі ширше застосовуються багатопроцесорні архітектури, особливо в галузях, де помилки можуть мати серйозні наслідки. Для таких систем надзвичайно важливо забезпечити високий рівень надійності, адже навіть незначний збій здатний спричинити значні фінансові втрати або поставити під загрозу людське життя. З метою підвищення стійкості до відмов використовуються механізми резервування та різноманітні методи реконфігурації. Саме вони забезпечують відмовостійкість багатопроцесорних систем (ВБС), дозволяючи їм продовжувати виконання поставлених функцій навіть у разі виходу з ладу окремих елементів.

Сучасні комп'ютерні комплекси значною мірою ґрунтуються на використанні відмовостійких багатопроцесорних систем, які призначені для розв'язання складних і відповідальних завдань. Їх основним призначенням є підтримання безперервного функціонування навіть за умов несправності окремих модулів. Особливо важливу роль такі системи відіграють у таких сферах:

- авіаційна та космічна техніка;
- енергетичні комплекси та електростанції;
- транспортна інфраструктура, зокрема залізничні системи;
- медичне обладнання та системи підтримки життєдіяльності.

Для характеристики ВБС часто використовується поняття «базова система», під яким розуміють систему, що зберігає працездатність після відмови визначеної кількості модулів. Такі системи позначаються як $K(m, n)$, де m характеризує максимально допустиму кількість відмов, які система може витримати без втрати функціональності, а n означає загальну кількість модулів.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		7

Одним із ключових завдань проєктування подібних систем є пошук оптимальної структури, яка забезпечувала б високий рівень надійності при мінімальних витратах на створення та експлуатацію. Для цього широко застосовується моделювання поведінки ВБС в потоці відмов, що дозволяє оцінювати показники надійності за допомогою статистичних експериментів над моделлю.

1.1.1. Основні характеристики ВБС

- Надійність – здатність системи зберігати працездатність при виникненні визначеної кількості відмов.
- Висока продуктивність – використання декількох процесорів забезпечує паралельне виконання значного обсягу обчислень.
- Масштабованість – архітектура ВБС дозволяє збільшувати кількість процесорів для підвищення продуктивності та рівня відмовостійкості.

Реконфігурація та резервування

Однією з ключових особливостей ВБС є можливість реконфігурації. Для забезпечення відмовостійкості використовуються дві базові стратегії.

Пасивна відмовостійкість

Передбачає створення такої архітектури системи, яка здатна підтримувати працездатність навіть за наявності несправних компонентів. Прикладом є системи з мажоритарним голосуванням, де правильне рішення визначається більшістю працездатних модулів.

Активна відмовостійкість

Передбачає виявлення несправностей, визначення їх місцезнаходження та автоматичний перерозподіл навантаження між працездатними елементами. Саме цей підхід найчастіше застосовується в сучасних ВБС, оскільки він дозволяє системі адаптуватися до змін свого технічного стану. Для реалізації такого механізму системи повинні підтримувати функції самодіагностики та постійного контролю стану своїх компонентів.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		8

1.1.2.Основні стани

Для відмовостійких багатопроцесорних систем характерні такі стани функціонування:

Нормальний стан

Усі процесори працюють у штатному режимі та виконують свої функції. У цьому випадку система забезпечує максимальну продуктивність і високий рівень надійності.

Стан збою

Унаслідок виникнення несправностей може відмовити один або декілька процесорів. Система повинна виявити проблему та вжити заходів для подальшого функціонування.

Стан відновлення

Після виявлення збою система переходить до виконання процедур відновлення та повернення до штатного режиму роботи.

Стан перенаправлення

У разі відмови окремих процесорів завдання автоматично передаються іншим працездатним вузлам. Хоча продуктивність при цьому може знижуватися, система зберігає свою функціональність і надійність.

Окрім зазначених станів, ВБС можуть перебувати в режимах моніторингу, тестування та налаштування конфігурації. Розуміння цих режимів є необхідною умовою для проєктування, експлуатації та забезпечення високої надійності таких систем.

Проблеми забезпечення відмовостійкості

Розроблення ВБС пов'язане з необхідністю знаходження компромісу між надійністю та продуктивністю. Реалізація механізмів резервування підвищує стійкість системи до відмов, але водночас ускладнює її архітектуру та збільшує обчислювальні й фінансові витрати.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		9

1.1.3.Практичне значення

Відмовостійкі багатопроцесорні системи застосовуються в тих сферах, де навіть незначна помилка може призвести до серйозних наслідків. До таких сфер належать банківські системи, атомна енергетика, авіаційна промисловість, медичне обладнання та інші критично важливі галузі. Саме тому розвиток і вдосконалення ВБС залишається одним із пріоритетних напрямів сучасної комп'ютерної інженерії.

У цьому розділі розглядаються базові поняття та принципи, що становлять основу подальшого дослідження методів моделювання та оптимізації відмовостійких багатопроцесорних систем.

1.1.4.Приклади застосування

1. Системи керування енергетичними об'єктами

ВБС використовуються для підтримки безперервної роботи диспетчерських центрів, контролю розподілу електроенергії, моніторингу стану мереж та оперативного реагування на аварійні ситуації.

Особливості:

- надзвичайно високі вимоги до надійності;
- необхідність роботи в режимі реального часу;
- використання резервних модулів для забезпечення безперервності функціонування.

2. Медичні діагностичні комплекси

Такі системи застосовуються в кардіомоніторах, апаратах штучної вентиляції легень, магнітно-резонансних томографах та інших медичних пристроях.

Особливості:

- критична важливість для життя пацієнтів;
- інтеграція з локальними та хмарними базами даних;
- здатність продовжувати роботу навіть при відмові окремих вузлів.

3. Центри обробки даних

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		10

ВБС забезпечують функціонування хмарних сервісів, систем зберігання інформації та інтернет-платформ.

Особливості:

- можливість одночасної обробки мільйонів запитів;
- застосування балансування навантаження та апаратного резервування;
- підтримка безперервної роботи у режимі 24/7.

4. Телекомунікаційні мережі

Використовуються для забезпечення передачі голосових, мультимедійних та інформаційних потоків між мільярдами користувачів.

Особливості:

- глобально розподілена архітектура;
- стійкість до втрати окремих вузлів і каналів зв'язку;
- використання ефективних алгоритмів маршрутизації та балансування навантаження.

5. Розподілені системи штучного інтелекту

До цієї категорії належать автономні транспортні засоби, робототехнічні комплекси та системи аналізу великих даних.

Особливості:

- децентралізована структура з підтримкою периферійних обчислень;
- виконання критично важливих завдань у реальному часі;
- можливість адаптації та самонавчання в нових умовах.

1.2 Існуючі методи та їх недоліки

1.2.1 Комбінаторні методи

Комбінаторні методи належать до класу аналітичних підходів, які базуються на дослідженні всіх можливих станів функціонування системи та визначенні ймовірностей переходів між ними. Основною метою цих методів є встановлення впливу окремих компонентів на загальну працездатність системи та визначення умов, за яких може настати її відмова.

До найбільш поширених комбінаторних методів належать:

- метод мінімальних шляхів і мінімальних перетинів;
- метод дерева відмов.

Комбінаторний підхід є одним із найпоширеніших засобів аналізу надійності технічних систем, оскільки дозволяє формалізувати структуру системи та оцінити її поведінку шляхом розгляду всіх можливих комбінацій станів елементів.

1.2.2 Метод мінімальних шляхів і мінімальних перетинів

У межах цього методу система подається у вигляді графа, де вершини відповідають окремим компонентам або вузлам, а ребра відображають зв'язки між ними. Мінімальний шлях являє собою найменший набір елементів, достатній для забезпечення працездатності системи. У свою чергу, мінімальний перетин визначається як мінімальна множина компонентів, одночасна відмова яких призводить до втрати працездатності всієї системи.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		12

Основні етапи застосування методу включають:

1. Формування графової моделі досліджуваної системи;
2. Визначення всіх мінімальних шляхів, що забезпечують функціонування системи;
3. Знаходження мінімальних перетинів;
4. Обчислення показників надійності на основі аналізу можливих комбінацій працездатних і непрацездатних станів елементів.

Застосування цього підходу дозволяє виявити найбільш критичні компоненти та оцінити їхній внесок у забезпечення загальної надійності системи.

1.2.3 Метод дерева відмов

Метод дерева відмов базується на побудові ієрархічної логічної структури, яка відображає причинно-наслідкові зв'язки між відмовами окремих елементів та відмовою системи в цілому.

У такій структурі верхня подія відповідає загальній відмові системи, тоді як нижні рівні описують несправності окремих компонентів або підсистем. Для формування дерева використовуються логічні оператори:

- оператор «АБО», який означає, що відмова системи настає за умови відмови хоча б одного з визначених елементів;
- оператор «І», який вказує, що відмова системи виникає лише за одночасної відмови кількох компонентів.

Побудоване дерево відмов дозволяє сформулювати логічні вирази для визначення ймовірності відмови системи, а також спрощує пошук критичних елементів структури.

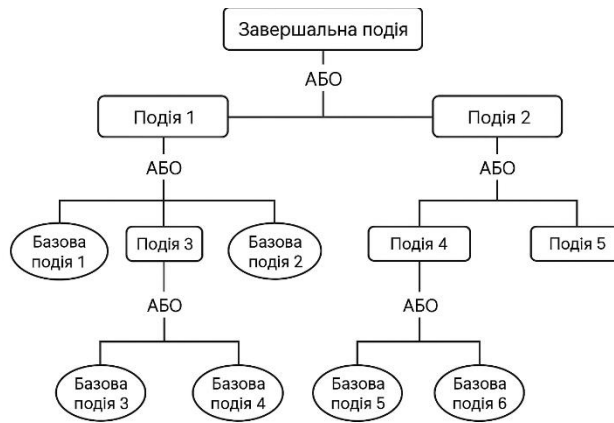


Рисунок 1 – Приклад діаграми дерева відмови

1.2.4 Марковські моделі

Марковські моделі широко використовуються для дослідження поведінки відмовостійких багатопроцесорних систем у процесі виникнення відмов та відновлення компонентів. Основою методу є застосування марковських процесів із дискретною множиною станів.

Особливістю марковського підходу є те, що майбутній стан системи визначається виключно її поточним станом і не залежить від попередньої історії функціонування. Завдяки цьому суттєво спрощується математичний опис процесів зміни станів системи.

Основними характеристиками марковських моделей є:

- представлення стану системи за допомогою вектора станів її компонентів;
- використання інтенсивностей відмов та відновлення для опису переходів між станами;
- можливість визначення показників надійності, зокрема середнього часу до відмови та ймовірності перебування системи в працездатному стані.

Процес побудови марковської моделі включає такі етапи:

1. Формування множини можливих станів системи;
2. Побудову матриці переходів між станами з урахуванням інтенсивностей відмов і відновлення;
3. Складання та розв'язання системи рівнянь Колмогорова;
4. Розрахунок показників надійності на основі отриманих імовірностей станів.

Перевагами марковського підходу є можливість врахування процесів відновлення та моделювання систем зі складною структурою. Водночас основним недоліком виступає значне збільшення кількості станів зі зростанням складності системи, що призводить до підвищення обчислювальних витрат.

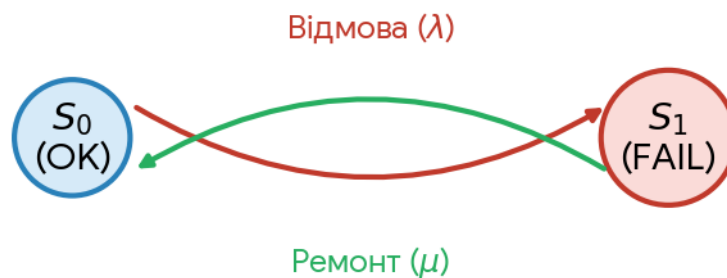


Рисунок 2 – Приклад Марковського методу

1.2.5 Метод функціональних блокових діаграм

Метод функціональних блокових діаграм передбачає подання системи у вигляді сукупності взаємопов'язаних функціональних блоків, кожен з яких відповідає окремому компоненту або підсистемі.

Залежно від структури системи блоки можуть бути з'єднані:

- послідовно;
- паралельно;
- комбінованим способом.

Для оцінювання надійності використовуються математичні співвідношення, що відповідають конкретному типу з'єднання. При послідовному з'єднанні надійність системи визначається добутком імовірностей безвідмовної роботи всіх елементів, тоді як для паралельного з'єднання враховується можливість резервування компонентів.

Основною перевагою цього методу є простота побудови моделей та наочність представлення структури системи. Разом із тим для систем зі складною топологією кількість необхідних розрахунків значно зростає, що обмежує ефективність його використання без застосування додаткових методів аналізу.

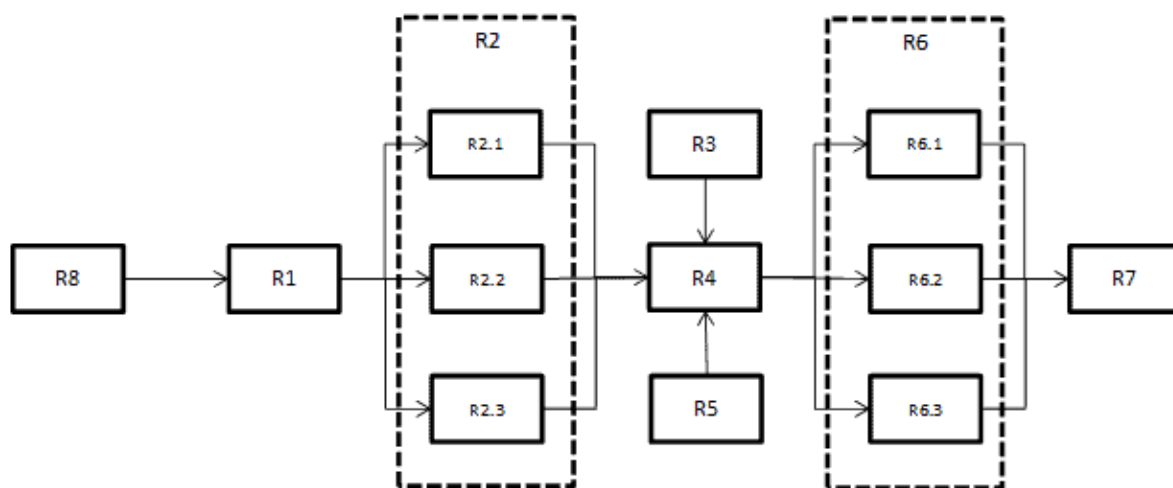


Рисунок 3 – Приклад методу блок-схеми

1.2.6 Метод Монте-Карло

Одним із найпоширеніших статистичних методів оцінювання надійності є метод Монте-Карло. Його сутність полягає у багаторазовому моделюванні роботи системи із застосуванням випадкових величин, що характеризують процеси відмов та відновлення елементів.

У межах даного підходу для кожного компонента генеруються випадкові значення часу до відмови відповідно до обраного закону розподілу, наприклад експоненціального, нормального або Вейбулла. Після цього визначається стан системи відповідно до прийнятого критерію працездатності.

Процедура моделювання включає:

1. Генерування випадкових параметрів для всіх елементів системи;
2. Визначення моментів виникнення відмов;
3. Аналіз працездатності системи на кожному етапі моделювання;
4. Повторення експерименту значну кількість разів;
5. Статистичну обробку отриманих результатів.

Метод Монте-Карло ефективно використовується для:

- аналізу складних багатопроцесорних систем;
- дослідження систем із нелінійними залежностями;
- оцінювання впливу резервування на надійність;
- врахування реальних умов експлуатації.

Основною перевагою методу є його універсальність, тоді як головним недоліком виступає потреба у значних обчислювальних ресурсах.

1.3 Комбіновані статистичні підходи

Для підвищення точності оцінювання надійності часто використовуються комбіновані статистичні методи, які поєднують результати аналітичних розрахунків і статистичного моделювання.

Застосування такого підходу дозволяє скоротити обсяг обчислень і водночас забезпечити достатню точність результатів. Найпоширенішими варіантами комбінування є:

1. Поєднання аналітичних методів із моделюванням Монте-Карло;
2. Розподіл системи на окремі підсистеми з подальшим статистичним аналізом кожної з них;
3. Використання аналітичних оцінок як початкових параметрів для статистичного моделювання.

Такий підхід особливо ефективний для великих багатопроцесорних систем зі складною структурою взаємозв'язків між компонентами.

1.3.1 Використання GL-моделей у комбінованих методах

GL-моделі є одним із перспективних інструментів моделювання складних відмовостійких систем. Їх використання дозволяє поєднати переваги графових структур із можливостями статистичного аналізу.

На етапі аналітичного дослідження GL-модель використовується для формування графа системи. Після побудови графової структури застосовується статистичне моделювання, що дозволяє дослідити поведінку системи за різних сценаріїв виникнення відмов. Особливо ефективними GL-моделі є для систем де функціонування одних елементів безпосередньо впливає на працездатність інших. У таких моделях можна аналізувати потоки ресурсів між вузлами та оцінювати наслідки відмов окремих компонентів для загального стану системи.

1.3.2 Практичне застосування комбінованих методів

Енергетичні системи

У системах керування енергетичними мережами комбіновані методи використовуються для аналізу надійності електричних мереж, підстанцій та диспетчерських комплексів.

Спочатку визначаються критичні маршрути передачі електроенергії, після чого моделюються можливі аварійні ситуації та їхній вплив на працездатність мережі.

Авіаційні та військові комплекси

У складних системах керування польотами комбіновані методи дозволяють визначити критичні вузли та оцінити ризики відмов за різних сценаріїв функціонування.

Статистичне моделювання використовується для аналізу аварійних режимів роботи та перевірки ефективності механізмів резервування.

Центри обробки даних

Для центрів обробки даних комбіновані підходи застосовуються під час оцінювання впливу відмов серверів, мережевого обладнання та систем зберігання інформації.

Моделювання дозволяє визначити необхідний рівень резервування та забезпечити безперервність надання сервісів.

Переваги та недоліки комбінованих методів

До переваг комбінованих методів належать:

- поєднання точності аналітичних методів із гнучкістю статистичного моделювання;
- можливість аналізу складних багаторівневих систем;
- врахування реальних умов експлуатації;
- зменшення обчислювальної складності завдяки декомпозиції системи;
- висока масштабованість підходу.

Недоліками є:

- складність інтеграції результатів різних методів;
- необхідність значних обчислювальних ресурсів;
- залежність кінцевих результатів від якості початкових даних;
- складність програмної реалізації.

1.4 Висновки до розділу

Проведений аналіз методів оцінювання надійності відмовостійких багатопроцесорних систем показав, що кожен із розглянутих підходів має власні переваги, обмеження та сфери ефективного застосування. Вибір конкретного методу залежить від складності структури системи, обсягу доступної інформації, необхідної точності результатів та наявних обчислювальних ресурсів.

Аналітичні методи забезпечують можливість отримання точних математичних залежностей між параметрами системи та показниками її надійності. Вони є найбільш ефективними для систем із відносно простою структурою та чітко

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		19

визначеними закономірностями функціонування. Однак зі збільшенням кількості компонентів та ускладненням зв'язків між ними використання аналітичних методів стає менш практичним через стрімке зростання складності математичних моделей.

Статистичні методи дозволяють враховувати випадковий характер виникнення відмов, різноманітні режими функціонування системи та реальні умови експлуатації. Вони характеризуються високою гнучкістю та універсальністю, проте потребують значних обчислювальних ресурсів і великої кількості експериментів для отримання статистично достовірних результатів.

Комбіновані методи поєднують сильні сторони аналітичних і статистичних підходів. Саме тому вони є найбільш перспективним напрямом аналізу складних багатопроцесорних систем. Використання таких методів дозволяє отримувати високоточні результати за прийнятних витрат обчислювальних ресурсів.

Подальший розвиток засобів оцінювання надійності пов'язаний зі створенням автоматизованих програмних комплексів, здатних адаптуватися до різних типів систем і враховувати особливості їхньої архітектури. Особливу актуальність мають методи аналізу великомасштабних розподілених багатопроцесорних структур.

2 GL-МОДЕЛЬ

2.1 Графо-логічна модель

2.1.1 Графо-логічні моделі відмовостійких багатопроцесорних систем

GL-моделі забезпечують комплексний підхід до опису зв'язності графової структури, яка відображає працездатний стан багатопроцесорної системи. У межах такого підходу кожне ребро графа характеризує важливий зв'язок між окремими компонентами системи, а його вилучення внаслідок відмови призводить до зміни загального стану функціонування системи.

2.1.2 Основні поняття GL-моделей

Вектор стану системи являє собою булевий вектор, елементи якого характеризують стан окремих процесорів. Значення «1» відповідає працездатному стану компонента, а значення «0» — його відмові. Наприклад, для вектора стану [1,1,0,1] несправним є лише третій процесор, тоді як решта компонентів продовжують функціонувати.

Реберні функції представляють собою логічні вирази, що визначають наявність або відсутність відповідного ребра в графі. Наприклад, для функції виду $f = x_1 \wedge x_2$ ребро буде існувати лише за умови, що обидва процесори x_1 та x_2 перебувають у працездатному стані.

Зв'язність графа є основним критерієм оцінювання працездатності системи. Якщо граф залишається зв'язним, система вважається працездатною. Порушення зв'язності графа свідчить про втрату працездатності всієї системи або її критичної частини.

2.1.3 Ефективність використання GL-моделей

Використання графо-логічних моделей надає широкі можливості для аналізу та дослідження багатопроцесорних систем. Зокрема, вони дозволяють:

- виконувати імітаційне моделювання процесів виникнення відмов;
- оцінювати показники надійності системи на різних етапах її життєвого циклу;
- здійснювати оптимізацію конфігурації системи ще до її практичного впровадження;
- визначати найбільш уразливі елементи архітектури;
- аналізувати вплив окремих компонентів на загальну відмовостійкість системи.

2.1.4 Недоліки та обмеження GL-моделей

Попри значні переваги, графо-логічні моделі мають і певні недоліки. До основних обмежень належать:

- висока обчислювальна складність під час моделювання великих систем;
- необхідність ретельного проєктування та налаштування реберних функцій;
- підвищена чутливість до похибок у вихідних параметрах моделі;
- збільшення складності аналізу зі зростанням кількості елементів системи.

2.1.5 GL-моделі як засіб моделювання надійності

Графо-логічні моделі вважаються одним із найбільш ефективних засобів дослідження надійності відмовостійких багатопроцесорних систем. Їх основна концепція полягає у поданні системи у вигляді графа, вершини якого відповідають процесорам або обчислювальним модулям, а ребра відображають інформаційні чи функціональні зв'язки між ними.

Стан кожного компонента задається за допомогою булевих змінних, які використовуються при формуванні реберних функцій. Завдяки цьому стає

можливим формалізований опис поведінки системи за різних комбінацій відмов компонентів.

2.1.6 GL-моделі поведінки відмовостійких систем у потоці відмов

Для моделювання поведінки відмовостійких багатопроцесорних систем в умовах виникнення відмов широко застосовуються графо-логічні моделі. Їхня структура базується на поєднанні теорії графів та апарату булевої алгебри.

Основою GL-моделі є неорієнтований граф, кожному ребру якого ставиться у відповідність певна булева функція. Ця функція може залежати від усіх або лише від окремих елементів вектора стану системи. Якщо значення функції стає рівним нулю, відповідне ребро автоматично виключається з графа.

Працездатність системи безпосередньо пов'язується зі зв'язністю графа моделі. Якщо для певного вектора станів граф залишається зв'язним, система продовжує виконувати свої функції. У випадку втрати зв'язності графа система вважається такою, що перейшла в стан відмови.

У загальному випадку структура графа та математичний опис реберних функцій можуть бути довільними. Особливий інтерес становлять моделі, побудовані на основі циклічних графів. Для таких структур процедура перевірки зв'язності є достатньо простою: граф зберігає зв'язність доти, доки з нього вилучено не більше одного ребра.

GL-моделі, що описують базові системи, прийнято називати базовими моделями та позначати як $K(m,n)$, де m визначає допустиму кількість відмов, а n — загальну кількість процесорних модулів. Моделі, що не відповідають зазначеним умовам, належать до класу небазових.

2.1.7 Особливості графо-логічних моделей

Графо-логічні моделі є універсальним інструментом аналізу та візуалізації інформації завдяки можливості відображення складних взаємозв'язків між об'єктами у вигляді вершин та ребер графа. У таких моделях вершини представляють окремі сутності або компоненти системи, а ребра відображають наявні між ними зв'язки. Завдяки своїй гнучкості GL-моделі широко використовуються в різних галузях науки та техніки, зокрема в інформатиці, біології, фізиці, економіці, соціології та теорії надійності.

До основних характеристик графо-логічних моделей належать:

- використання графічного представлення інформації;
- підтримка різних типів вершин та ребер;
- можливість моделювання складних взаємозв'язків;
- адаптивність до змін структури системи;
- підтримка динамічних сценаріїв функціонування.

Графічне представлення значно полегшує аналіз системи та підвищує наочність отриманих результатів. Можливість використання різних типів вузлів і зв'язків дозволяє враховувати особливості конкретної предметної області та створювати універсальні моделі для дослідження складних систем.

Однією з важливих проблем під час побудови графо-логічних моделей є оптимізація внутрішніх зв'язків графа. Надмірна кількість перетинів ребер може суттєво ускладнювати сприйняття структури моделі та знижувати ефективність її аналізу. Тому задачі мінімізації кількості перетинів і раціонального розміщення елементів графа мають важливе практичне значення.

2.1.8 GL-моделі та підходи до їх побудови

Графо-логічні моделі поєднують у собі властивості графових структур і булевих функцій. Вони реалізуються у вигляді неорієнтованих графів, де кожне ребро асоціюється з певною булевою функцією, яка залежить від елементів вектора стану системи.

Якщо значення відповідної функції дорівнює нулю, ребро видаляється з графа. Таким чином, зв'язність графа виступає критерієм працездатності системи: її збереження означає можливість подальшого функціонування, а втрата — виникнення відмови.

Для побудови GL-моделей розроблено декілька алгоритмів, які відрізняються принципами формування графа та реберних функцій. Одним із найбільш поширених є метод МВР (метод мінімуму втрачених ребер), який базується на використанні циклічних графів.

Основною перевагою цього методу є простота перевірки зв'язності графа та мінімізація кількості втрачених ребер для станів системи, що містять $m+1$ відмов. Саме тому метод МВР часто використовується як базовий підхід у дослідженнях, пов'язаних із побудовою GL-моделей.

2.1.9 Методи модифікації графо-логічних моделей

Методи побудови GL-моделей дозволяють формувати моделі базових систем, здатних зберігати працездатність доти, доки кількість несправних процесорів не перевищить встановленого порогу m . Такі системи прийнято називати базовими.

Однак реальні системи керування та обробки інформації часто мають складнішу поведінку. У багатьох випадках відмова однієї групи з m процесорів може спричинити повну втрату працездатності, тоді як відмова іншої групи тієї ж потужності не впливатиме на функціонування системи.

Для адекватного відображення таких особливостей необхідне модифікування базових GL-моделей. Внесення змін до структури моделей дозволяє врахувати специфіку роботи системи та забезпечити більш точне відображення її поведінки в потоці відмов.

У загальному випадку методи модифікації графо-логічних моделей поділяються на три основні групи:

1. Методи, що базуються на зміні структури графа;
2. Методи, засновані на модифікації реберних функцій;
3. Комбіновані методи, які поєднують обидва підходи.

Перша група передбачає додавання нових ребер і відповідних їм функцій до графової структури моделі. Такі методи є достатньо добре дослідженими, а задача довільної модифікації моделей для них уже розв'язана. Водночас додавання нових ребер змінює структуру графа, внаслідок чого він перестає бути циклічним, що ускладнює процедуру перевірки зв'язності.

Методи другої групи спрямовані на зміну лише логічних виразів реберних функцій без втручання у структуру графа. Завдяки цьому зберігається початкова топологія моделі та простота аналізу зв'язності. Проте ці методи наразі залишаються менш дослідженими, а задача довільної модифікації моделей для них ще не має універсального розв'язання.

Третя група методів базується на одночасній зміні структури графа та реберних функцій. Незважаючи на недостатній рівень дослідження, саме цей підхід вважається одним із найбільш перспективних, оскільки дозволяє поєднати переваги двох попередніх груп і забезпечити високу гнучкість під час моделювання складних відмовостійких систем.

2.2 Базова GL-модель

Базовою називають відмовостійку багатопроцесорну систему, яка залишається працездатною для всіх можливих векторів стану, де кількість несправних компонентів не перевищує певного фіксованого значення k . Іншими словами, така система здатна повноцінно виконувати свої функції навіть за наявності обмеженої кількості відмов. Якщо ж число несправних елементів досягає або перевищує встановлений поріг, система може продовжувати функціонувати, однак її можливості та ефективність можуть бути частково обмежені.

Максимальна кількість компонентів, відмова яких не призводить до втрати працездатності системи, називається кратністю відмовостійкості. Базову відмовостійку багатопроцесорну систему, що складається з n елементів і здатна витримувати відмову не більше ніж m компонентів (де $0 \leq m \leq n$), прийнято позначати як $K(m,n)$.

Водночас більшість реальних багатопроцесорних систем не належать до класу базових. У практичних умовах система може втратити працездатність через певну комбінацію з k відмов, але залишатися функціональною за наявності іншої конфігурації відмов, навіть якщо кількість несправних компонентів є більшою та дорівнює s , де $s > k$. Така особливість свідчить про те, що для дослідження реальних систем недостатньо використовувати лише моделі базових ВБС. Натомість виникає необхідність створення моделей, здатних відображати поведінку небазових систем у процесі виникнення та накопичення відмов.

Розробка подібних моделей передбачає вирішення низки задач, пов'язаних із модифікацією структури графів та перетворенням реберних функцій графологічних моделей. Це необхідно для більш точного опису особливостей функціонування системи та підвищення ефективності подальшого моделювання.

Небазові відмовостійкі багатопроцесорні системи характеризуються тим,

що їх реакція на відмови окремих компонентів не є однозначною. Це пояснюється наявністю в їх структурі критично важливих процесорів, вихід з ладу яких суттєво впливає на працездатність усієї системи, а також менш значущих елементів, відмова яких має незначний вплив на загальне функціонування. У результаті однакова кількість відмов може призводити до різних наслідків залежно від того, які саме компоненти вийшли з ладу. Саме ця особливість робить аналіз небазових ВБС більш складним і водночас більш наближеним до реальних умов експлуатації сучасних обчислювальних систем.

2.3 Побудова базової моделі

GL- модель базової ВБС може бути сформована, зокрема, за методом, наведеним у МВР. Модель $K(m, n)$, для якої $n \geq 1$ та $1 \leq m \leq n$, що будується відповідно до МВР, ґрунтується на циклічному графі з $\phi(n, m)$ ребрами, де $\phi(n, m) = n - m + 1$. Додатково, однією з характеристик таких моделей є те, що для будь якого вектора стану, який містить l нульових компонентів, модель втрачає $\psi(m, l)$ ребер, причому $\psi(m, l) = \{0, \text{якщо } l < m; l - m + 1, \text{якщо } l \geq m.\}$

Подібні GL-моделі називаються МВР-моделями (мінімуму втрачених ребер), оскільки у випадку векторів стану з $m + 1$ нульовими компонентами вони втрачають мінімально можливу кількість ребер, видалення яких призводить до порушення зв'язності графа (а саме два ребра). Формування виразів реберних функцій МВР-моделі виконується рекурсивно на базі виразів реберних функцій допоміжних МВР-моделей меншої розмірності. Вектор стану системи поділяється на два непорожні підвектори розмірів n_1 та n_2 , де $n_1 + n_2 = n$, які застосовуються як вхідні дані для відповідних допоміжних моделей. Загалом значення n_1 і n_2 можуть бути довільними, однак у практичних застосуваннях поділ здебільшого здійснюється на дві рівні або близькі за розміром частини. Множина реберних функцій моделі $K(m, n)$ утворюється як об'єднання трьох підмножин: - множини реберних функцій допоміжної МВР-моделі $K_1(m, n_1)$ за умови, що $n_1 \geq m$; - множини функцій вигляду $f = \Lambda(f_1 \in F_1) f_1 \vee \Lambda(f_2 \in F_2) f_2$,

де $F1$ є множиною реберних функцій допоміжної МВР-моделі $K1(m - i, n1)$, а $F2$ — множиною реберних функцій допоміжної МВР-моделі $K2(i, n2)$, при $i = a, a + 1, \dots, b$, де $a = \max(1, m - n1)$, $b = \min(m - 1, n2)$, за умови, що $m \geq 2$; - множини реберних функцій допоміжної МВР-моделі $K2(m, n2)$ за умови, що $n2 \geq m$. Рекурсивна процедура побудови триває до досягнення граничного рівня — моделей типу $K(1, 1)$, які містять одну реберну функцію виду $f = x_i$, де x_i є відповідним компонентом вектора стану системи.

Запропонований підхід є ефективним для побудови GL-моделей базових систем. Водночас задача побудови GL-моделей для небазових ВБС, зокрема тих, поведінка яких визначається складнішими комбінаціями станів процесорів, потребує окремого аналізу.

2.4 Небазова модель

Небазові графо-логічні моделі застосовуються для дослідження відмовостійких систем, у яких працездатність визначається не лише загальною кількістю несправних компонентів, а й їх конкретним розташуванням або певними комбінаціями відмов. Такі моделі особливо ефективні для аналізу складних багатопроцесорних систем із неоднорідною структурою, де окремі елементи мають різний ступінь впливу на функціонування всієї системи.

Особливістю небазових моделей є врахування індивідуальних характеристик компонентів. Наприклад, деякі вузли можуть виконувати критично важливі функції та мати вирішальне значення для працездатності системи, тоді як відмова інших елементів впливає на її роботу значно менше. У зв'язку з цим однакова кількість відмов може призводити до різних наслідків залежно від того, які саме компоненти вийшли з ладу.

На відміну від базових моделей, небазові GL-моделі дозволяють враховувати різноманітні сценарії виникнення несправностей та оцінювати вплив кожної конкретної комбінації відмов на загальний стан системи. Це забезпечує більш реалістичне відображення процесів функціонування складних

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		29

обчислювальних комплексів.

Для моделювання небазових систем стандартних графо-логічних моделей часто недостатньо. Тому при їх побудові використовуються модифіковані реберні функції, які враховують специфічні особливості архітектури системи, умови її експлуатації та різний рівень важливості окремих компонентів.

Переваги та недоліки базових і небазових моделей

Базові GL-моделі характеризуються відносною простотою побудови та аналізу. Вони добре підходять для систем із однорідною структурою, де всі компоненти мають однаковий рівень значущості та відмовостійкості. Крім того, такі моделі потребують менших обчислювальних витрат під час проведення аналізу. Водночас їх використання є обмеженим у випадку складних систем із неоднаковими умовами функціонування елементів.

Таблиця 1 - Порівняння небазових та базових моделей

Параметр	Небазова модель	Базова модель
Реберні функції	Модифіковані, складніші	Простішої структури
Стійкість до відмов	Вагомість певних вузлів відрізняється	Однакова для всіх вузлів однаково
Структура графа	Нерівномірна	Однорідна
Обчислювальна складність	Вища	Нижча

Небазові графо-логічні моделі забезпечують значно вищу гнучкість і дозволяють більш точно описувати реальні системи. Вони здатні враховувати специфіку окремих вузлів, різноманітні сценарії відмов та особливості взаємодії між компонентами. Однак підвищення точності супроводжується зростанням складності побудови моделей та збільшенням обсягу необхідних обчислень.

Сфери застосування

Базові графо-логічні моделі переважно використовуються для аналізу систем із рівномірною структурою та однаковими характеристиками всіх компонентів. Прикладом можуть бути обчислювальні кластери, що складаються з однотипних процесорних модулів. Для систем типу $K(m,n)$ такі моделі дозволяють оцінювати збереження зв'язності графа за умови відмови визначеної кількості компонентів.

Небазові моделі застосовуються для дослідження складних систем, у яких окремі елементи мають підвищену важливість. Наприклад, у хмарних обчислювальних інфраструктурах центральні сервери можуть бути значно критичнішими за допоміжні вузли. У таких випадках реберні функції будуються з урахуванням пріоритетності компонентів та їх індивідуальних характеристик надійності.

Особливості побудови небазових GL-моделей

Однією з ключових особливостей небазових моделей є використання спеціально розроблених реберних функцій. Для кожного ребра графа можуть визначатися унікальні булеві вирази, які враховують роль відповідних компонентів у забезпеченні працездатності системи.

Для зменшення обчислювальної складності також широко застосовуються комбінаторні методи, що передбачають поділ великого графа на окремі підграфи. Такий підхід дозволяє спростити процес аналізу та підвищити ефективність моделювання складних багатопроцесорних систем.

2.5 Побудова небазової моделі

Розглянемо небазову ВБС, яка є працездатною лише за умови одночасного виконання деякого набору умов $C_1, C_2, \dots, C_k$. Для кожної з цих умов C_i певним чином може бути сформована відповідна GL-модель M_i (далі називатимемо їх допоміжними моделями). Отже, виконання умови C_i визначається зв'язністю графа моделі M_i .

Оскільки система функціонує лише тоді, коли виконуються всі умови C_i ,

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		31

граф GL-моделі М цієї ВБС повинен залишатися зв'язним тоді, коли зв'язними є графи всіх моделей M_i , і навпаки — втратити зв'язність, якщо хоча б один із графів моделей M_i стає незв'язним.

Розглянемо два довільні графи G_1 та G_2 . Нехай граф G_1 має вершини $\alpha_1, \alpha_2, \dots, \alpha_{n1}$, а граф G_2 — вершини $\beta_1, \beta_2, \dots, \beta_{n2}$. Виберемо довільні вершини α_i та β_j із графів G_1 та G_2 відповідно. Виконаємо об'єднання графів G_1 і G_2 через ці вершини, тобто зіллємо α_i та β_j в одну вершину, а всі інші вершини та ребра обох графів перенесемо без змін. У результаті такого об'єднання отримаємо граф G .

Якщо граф G_1 є зв'язним, це означає, що між будь-якою парою його вершин існує шлях. Якщо ж граф G_1 незв'язний, то в ньому знайдеться хоча б одна пара вершин, між якими відсутній шлях. Аналогічно, зв'язність графа G_2 гарантує наявність шляху між будь-якими його вершинами, тоді як незв'язність означає існування хоча б однієї пари вершин без шляху між ними.

Покажемо, що граф G залишається зв'язним, якщо обидва графи G_1 і G_2 є зв'язними, і стає незв'язним, якщо хоча б один із них втрачає зв'язність.

Припустимо, що графи G_1 і G_2 зв'язні. Виберемо довільну пару вершин графа G . Можливі три варіанти.

1. Обидві вершини належать множині $\{\alpha_1, \alpha_2, \dots, \alpha_{n1}\}$. Тоді шлях між ними існує завдяки зв'язності графа G_1 .

2. Обидві вершини належать множині $\{\beta_1, \beta_2, \dots, \beta_{n2}\}$. У цьому випадку шлях забезпечується зв'язністю графа G_2 .

3. Одна вершина (позначимо її α_k) належить першій множині, а інша (позначимо β_l) — другій. Із зв'язності графа G_1 випливає існування шляху між α_k та α_i , а із зв'язності G_2 — між β_j та β_l . Оскільки в графі G вершини α_i і β_j об'єднані, то існує шлях і між вершинами α_k та β_l .

Отже, якщо графи G_1 і G_2 є зв'язними, то і граф G також буде зв'язним.

Тепер доведемо, що граф G стане незв'язним, якщо хоча б один із графів G_1 або G_2 є незв'язним.

Нехай граф G_1 незв'язний. Тоді існує хоча б одна пара вершин із множини $\{\alpha_1, \alpha_2, \dots, \alpha_{n1}\}$, між якими немає шляху. Очевидно, що в такому разі принаймні одна з цих вершин не має шляху до вершини α_i (інакше через α_i шлях між ними існував би). Відповідно, у графі G не буде шляху від цієї вершини до жодної вершини з множини $\{\beta_1, \beta_2, \dots, \beta_{n2}\}$, оскільки з'єднання було здійснено лише між α_i і β_j , без додавання нових ребер. Отже, граф G є незв'язним. Аналогічне міркування справджується і для випадку, коли незв'язним є граф G_2 .

Описану процедуру об'єднання можна узагальнити на будь-яку кількість графів. Використаємо її для об'єднання графів допоміжних моделей $M_1, M_2, \dots, M_k$.

Очевидно, що отримана таким способом GL-модель M відображатиме працездатність системи тоді і лише тоді, коли виконуються всі умови $C_1, C_2, \dots, C_k$, тобто коли графи всіх моделей $M_1, M_2, \dots, M_k$ є зв'язними. Дійсно, з одного боку зв'язність кожного з цих графів забезпечує зв'язність графа моделі M . З іншого боку, об'єднання графів через спільні вершини не створює додаткових шляхів, які могли б зберегти зв'язність графа M , якщо хоча б один із графів $M_1, M_2, \dots, M_k$ її втрачає.

Крім того, варто зазначити, що цей метод дозволяє довільно обирати як порядок об'єднання моделей, так і вершини, через які відбувається злиття графів. Таким чином, для однієї й тієї ж ВБС можна отримати значну кількість різних варіантів GL-моделі. Це, зокрема, дає змогу сформувати більш зручну конфігурацію графа, наприклад, для прискорення перевірки зв'язності або для спрощення подальших змін моделі за необхідності.

2.6 Модифікація небазової моделі

Модифікація небазових графо-логічних моделей

Одним із важливих напрямів дослідження графо-логічних моделей є їх адаптація до опису реальних технічних систем. На практиці більшість складних обчислювальних і керуючих систем не належать до класу базових, тому для адекватного відображення їх поведінки необхідно використовувати небазові GL-моделі. Такі моделі характеризуються складнішою структурою та більш різноманітними сценаріями функціонування в умовах виникнення відмов, що потребує застосування спеціальних підходів до їх побудови та модифікації.

На відміну від базових моделей, небазові GL-моделі повинні враховувати особливості конкретної системи, її архітектуру та специфіку взаємодії між компонентами. З цієї причини основою їх формування зазвичай виступає вже побудована базова модель, яка надалі піддається певним змінам для досягнення необхідної відповідності реальному об'єкту дослідження.

Процес переходу від базової моделі до небазової може здійснюватися декількома способами. Перший підхід передбачає зміну структури графа моделі. У цьому випадку до графа можуть додаватися нові ребра, вилучатися існуючі або змінюватися характер зв'язків між вершинами. Така модифікація дозволяє точніше відобразити реальні взаємозв'язки між елементами системи та врахувати особливості її функціонування.

Іншим підходом є коригування реберних функцій без зміни самої структури графа. Зміна логічних виразів, які визначають умови існування ребер, дає можливість змінювати поведінку моделі в потоці відмов, не порушуючи початкову топологію графа. Саме цей підхід становить найбільший інтерес, оскільки дозволяє зберегти переваги базових моделей і водночас забезпечити необхідну гнучкість під час моделювання складних систем.

У межах даного дослідження основна увага приділяється саме модифікації реберних функцій. Методи, розроблені для базових GL-моделей, можуть бути

адаптовані та розширені для роботи з небазовими структурами, що значно збільшує сферу їх практичного застосування.

Вплив модифікації на поведінку моделі

У результаті перетворення базової моделі типу $K(m,n)$ на небазову модель $K'(m,n)$ змінюється її реакція на окремі стани системи. Сукупність усіх можливих векторів станів при цьому доцільно поділити на декілька категорій залежно від характеру змін у поведінці моделі.

До першої категорії належать вектори станів, для яких поведінка модифікованої та базової моделей залишається однаковою. У цих випадках внесені зміни не впливають на оцінку працездатності системи, тому результати функціонування обох моделей збігаються.

Другу категорію утворюють так звані посилені вектори станів. Для них модифікована модель продовжує вважати систему працездатною, хоча базова модель уже класифікує відповідний стан як відмову. Наявність таких векторів свідчить про підвищення стійкості моделі до окремих комбінацій відмов.

Третю категорію становлять послаблені вектори. У цьому випадку ситуація є протилежною: базова модель визначає систему як працездатну, тоді як модифікована модель фіксує втрату працездатності. Такі стани дозволяють врахувати додаткові ризики та критичні режими роботи системи.

Саме за рахунок використання посилених і послаблених векторів досягається адаптація моделі до особливостей реальної системи. Посилення окремих станів дозволяє враховувати додаткові можливості резервування та реконфігурації, тоді як послаблення забезпечує більш суворий контроль за критичними режимами функціонування.

Переваги використання модифікованих небазових моделей

Застосування методів модифікації небазових GL-моделей надає низку суттєвих переваг під час аналізу відмовостійких систем.

Перш за все, модифіковані моделі дозволяють значно підвищити рівень відповідності між математичним описом та реальною системою. Завдяки

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		35

коригуванню реберних функцій можна враховувати специфічні особливості функціонування конкретного об'єкта без необхідності повної перебудови моделі.

Крім того, модифікація забезпечує можливість дослідження складних потоків відмов і нестандартних сценаріїв роботи системи. Це сприяє більш точному прогнозуванню поведінки системи в умовах експлуатації та підвищує якість оцінювання її надійності.

Ще однією перевагою є можливість оптимізації структури моделі. У деяких випадках модифікація дозволяє усунути надлишкові елементи базової моделі, а в інших — навпаки, додати необхідні залежності між компонентами для точнішого відображення реальних процесів.

Таким чином, використання методів модифікації небазових графо-логічних моделей дозволяє отримати більш реалістичний опис поведінки відмовостійких багатопроцесорних систем, підвищити точність оцінювання їх надійності та розширити можливості аналізу складних процесів функціонування в умовах виникнення відмов.

2.6 Перевірка коректності моделі

Схема проведення експериментального дослідження наведена на рисунку 3. Під час виконання експерименту згенеровані вектори станів системи одночасно подаються на два незалежні модулі: побудовану графо-логічну модель та модуль програмної імітації функціонування системи в умовах виникнення відмов. Після отримання результатів від обох модулів виконується їх порівняння. У випадку виявлення розбіжностей відповідна інформація фіксується у звіті. Після завершення перевірки для всіх досліджуваних станів можна зробити висновок щодо адекватності побудованої моделі. Якщо результати GL-моделі повністю збігаються з результатами симуляції для кожного перевіреного вектора, модель вважається такою, що коректно відображає поведінку системи в потоці відмов.

Для повної перевірки відповідності моделі необхідно проаналізувати всі можливі стани системи, тобто перебрати всі булеві вектори довжиною, що дорівнює кількості процесорів. Однак зі збільшенням розмірності системи кількість можливих комбінацій стрімко зростає, тому для багатопроцесорних систем із десятками процесорів повний перебір стає практично неможливим. У таких випадках застосовується перевірка на випадково сформованій множині векторів станів, що дозволяє оцінити коректність моделі з певним рівнем статистичної достовірності.

Після надходження вектора стану до модуля GL-моделі спочатку обчислюються значення всіх реберних функцій. Далі виконується модифікація структури графа шляхом видалення ребер, для яких відповідні функції набули нульового значення. Після оновлення графа здійснюється перевірка його зв'язності. Саме зв'язність графа є критерієм працездатності системи: якщо граф залишається зв'язним, система вважається працездатною, а якщо зв'язність порушується — система переходить у непрацездатний стан.

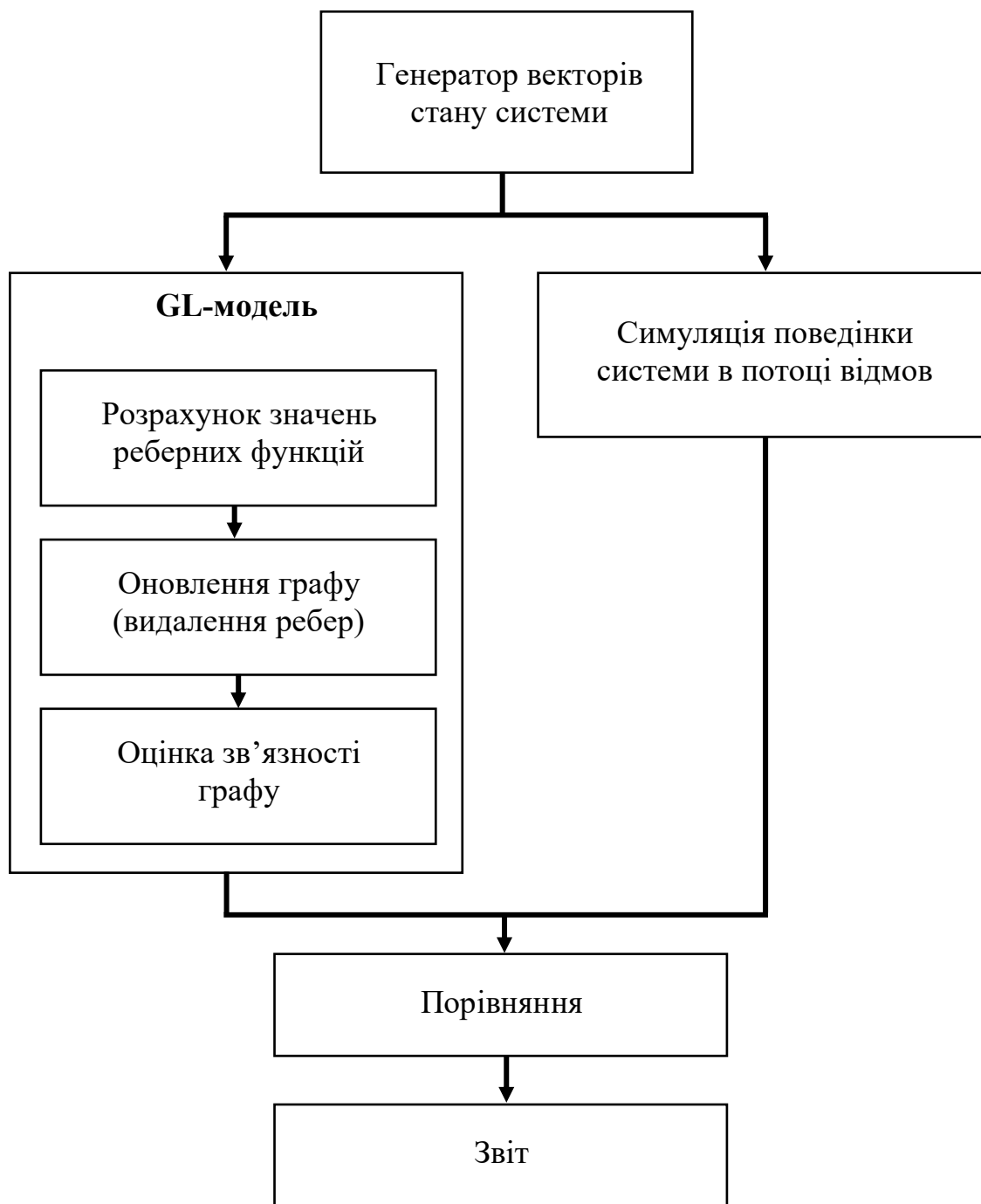


Рисунок 4 - Схема експерименту

На відміну від GL-моделі, модуль симуляції визначає працездатність системи без використання графового подання. Оскільки логіка функціонування системи відома заздалегідь, її поведінка може бути описана програмними засобами за допомогою умовних операторів, логічних виразів та математичних операцій. Для реалізації таких перевірок можуть використовуватися параметризовані алгоритми або спеціальні сценарії, створені засобами скриптових мов програмування. Результатом роботи цього модуля є визначення

факту працездатності або непрацездатності системи для заданого набору станів процесорів.

Модуль порівняння аналізує результати, отримані від GL-моделі та симулятора. Якщо обидва модулі повертають однаковий результат, вважається, що модель правильно відображає поведінку системи для відповідного вектора стану. У разі виникнення розбіжностей можна зробити висновок про невідповідність між моделлю та реальною логікою функціонування системи, за умови відсутності помилок у реалізації допоміжних модулів.

Усі випадки невідповідності реєструються у звіті із зазначенням вектора стану системи, результату, отриманого за допомогою графо-логічної моделі, та результату, сформованого модулем симуляції. Якщо після завершення перевірки звіт не містить жодного запису, це свідчить про коректність побудованої моделі. У разі наявності записів дослідник отримує можливість провести детальний аналіз проблемних ситуацій та визначити причини виникнення розбіжностей.

Для підтвердження ефективності запропонованого алгоритму побудови графо-логічних моделей небазових відмовостійких багатопроцесорних систем доцільно провести серію експериментів для значної кількості різних конфігурацій систем. Для кожної конфігурації необхідно сформулювати відповідну GL-модель та налаштувати модуль симуляції відповідно до характеристик досліджуваної системи. Виконання таких операцій вручну є надзвичайно трудомістким, тому доцільним є використання автоматизованого підходу.

Схема автоматизованого проведення експериментів для випадково сформованих систем наведена на рисунку 5. Такий підхід дозволяє виконувати перевірку великої кількості різноманітних систем без безпосередньої участі дослідника. Після завершення серії випробувань формується підсумковий звіт, який містить інформацію про всі виявлені невідповідності між результатами графо-логічного моделювання та програмної симуляції. Крім того, звіт дозволяє визначити характер помилки (хибнопозитивна чи хибнонегативна) та умови її виникнення.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		39



Рисунок 5 - Схема проведення експериментів із випадковими параметрами ВБС

Відповідно до запропонованої схеми генератор випадкових параметрів створює набори характеристик системи для чергового експерименту. До таких характеристик належать кількість процесорів n , число рівнів відмовостійкості k , значення показників відмовостійкості $m_1, m_2, \dots, m_k$, а також логічні умови, що визначають відповідний рівень відмовостійкості. Для забезпечення гнучкості експериментальних досліджень параметри генерації повинні налаштовуватися користувачем. Також доцільно вести облік уже використаних наборів параметрів для запобігання повторному тестуванню однакових конфігурацій.

На основі сформованих параметрів автоматично будується графо-логічна модель згідно з алгоритмом, описаним у попередньому розділі. Одночасно

виконується налаштування модуля симуляції: генерується відповідний програмний сценарій або встановлюються необхідні параметри його роботи.

Сформовані модель і симулятор використовуються для проведення окремого експерименту. При цьому модулю генерації випадкових векторів стану передається кількість процесорів системи, що визначає довжину сформованих тестових наборів.

Після завершення всіх перевірок результати окремих експериментів об'єднуються у єдиний звіт. У ньому зазначаються параметри систем, для яких були зафіксовані розбіжності, а також детальна інформація про кожен випадок невідповідності. Аналізуючи цей звіт, дослідник може визначити, для яких конфігурацій систем виникали помилки та за яких комбінацій станів процесорів вони проявлялися. Якщо ж після виконання всіх експериментів звіт не містить жодного запису, це свідчить про те, що графо-логічні моделі, побудовані за запропонованим алгоритмом, коректно відображають поведінку відповідних відмовостійких багатопроцесорних систем у потоці відмов.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		41

3 РОБОТА ПРОГРАМИ

3.1 Розробка програми

Для програми мовою програми був обраний python адже він є високорівневою мовою програмування загального призначення, яка здобула широку популярність завдяки простоті використання, зрозумілому синтаксису та значним можливостям для розв'язання різноманітних прикладних і наукових задач. Завдяки своїм особливостям ця мова активно застосовується у сфері наукових досліджень, автоматизації процесів, аналізу даних, машинного навчання та розробки складних програмних систем.

3.1.1 Переваги Python

Простота синтаксису Однією з головних переваг Python є його лаконічний і зрозумілий синтаксис. Код, написаний цією мовою, легко читається та підтримується, що дозволяє швидко реалізовувати алгоритми навіть без використання спеціалізованих зовнішніх бібліотек. Це особливо важливо під час створення дослідницьких програмних продуктів, де основна увага приділяється алгоритмічній логіці, а не технічним особливостям реалізації.

Гнучкість використання Python може ефективно застосовуватися для широкого спектра задач, серед яких обробка даних, математичні обчислення, моделювання систем, робота з графовими структурами та створення алгоритмів різної складності. Базові можливості мови дозволяють реалізувати значну частину необхідного функціоналу без залучення сторонніх інструментів.

Платформна незалежність Програмне забезпечення, розроблене мовою Python, може працювати практично на будь-якій сучасній операційній системі, включаючи Windows, Linux та macOS. Це забезпечує високу мобільність програмного продукту та спрощує його подальше впровадження й використання.

Висока швидкість розробки Завдяки простоті конструкцій і мінімальній кількості службового коду Python дозволяє значно скоротити час створення

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		42

програмних рішень. Це особливо актуально під час побудови прототипів та експериментальних моделей, зокрема систем моделювання GL-моделей.

Зручність роботи зі структурами даних Мова надає широкий набір вбудованих структур даних, серед яких списки, словники, множини та кортежі. Вони можуть ефективно використовуватися для представлення графів, опису зв'язків між елементами системи та реалізації алгоритмів їхньої обробки.

Можливість оптимізації реалізації У разі потреби Python дозволяє реалізовувати алгоритми з детальним контролем над процесом обчислень. Це дає змогу створювати власні механізми обробки даних без необхідності використання сторонніх програмних бібліотек, забезпечуючи прозорість та керованість усіх етапів виконання алгоритмів.

3.1.2 Недоліки Python

Порівняно невисока швидкодія Python належить до інтерпретованих мов програмування, тому швидкість виконання програм зазвичай поступається рішенням, створеним мовами C++, C# або Java. Проте для більшості задач цей недолік може бути компенсований застосуванням ефективних алгоритмів та раціональною організацією обчислень.

Підвищене споживання пам'яті Порівняно з низькорівневими мовами програмування Python використовує більше оперативної пам'яті для зберігання об'єктів та службових структур. У великих проєктах це може впливати на продуктивність системи та вимагати додаткової оптимізації.

Обмеження багатопотокового виконання Через наявність механізму Global Interpreter Lock (GIL) Python має певні особливості при реалізації багатопотокових обчислень. Однак для задач, що виконуються послідовно або не потребують інтенсивного паралелізму, дане обмеження не є критичним.

3.1.3 Обґрунтування вибору Python

Вибір Python для реалізації програмного комплексу обумовлений низкою факторів. Насамперед мова дозволяє реалізувати всі необхідні алгоритми побудови, модифікації та аналізу GL-моделей за допомогою стандартних засобів без залучення додаткових бібліотек.

Важливою перевагою є також простота реалізації алгоритмів. Завдяки зрозумілій структурі коду значно спрощується створення механізмів побудови графів, формування реберних функцій та аналізу зв'язності моделей.

Крім того, Python забезпечує високу зручність супроводу програмного забезпечення. Код легко читається, редагується та розширюється, що особливо важливо для проєктів, які передбачають подальший розвиток функціональних можливостей.

Ще одним аргументом на користь Python є швидкість створення прототипів. Це дозволяє оперативно реалізовувати нові підходи до побудови GL-моделей та проводити їх експериментальну перевірку.

Універсальність мови дає можливість одночасно реалізовувати моделі, тестувати алгоритми та аналізувати результати їх роботи в межах єдиного програмного середовища без необхідності використання додаткових технологічних платформ.

Таким чином, вибір Python для створення програмного комплексу побудови GL-моделей із симетричним резервуванням є цілком обґрунтованим. Поєднання простоти використання, гнучкості, швидкості розробки та достатньої функціональності дозволяє створити ефективне програмне рішення без залучення зовнішніх бібліотек і складних програмних засобів.

3.2 Методи та принципи, використані під час розробки

Об'єктно-орієнтоване програмування (ООП) є сучасною парадигмою розробки програмного забезпечення, яка базується на використанні класів та об'єктів. У межах цього підходу дані та засоби їх обробки об'єднуються в єдині програмні сутності, що забезпечує високу структурованість програмного коду та спрощує процес його подальшої підтримки.

Основною ідеєю ООП є моделювання реальних об'єктів предметної області за допомогою програмних класів. Кожен об'єкт містить власний набір характеристик та операцій, що дозволяє створювати логічно завершені та незалежні компоненти програмної системи.

3.2.1 Інкапсуляція

Інкапсуляція передбачає приховування внутрішньої реалізації об'єкта від зовнішнього середовища. Доступними залишаються лише ті методи та властивості, які необхідні для взаємодії з об'єктом. Такий підхід забезпечує захист даних від неконтрольованих змін і підвищує надійність програмної системи.

3.2.2 Абстракція

Абстракція дозволяє виділити найбільш важливі характеристики об'єкта та приховати другорядні деталі його реалізації. Завдяки цьому взаємодія з програмними компонентами стає простішою та більш зрозумілою для розробника.

3.2.3 Наслідування

Наслідування забезпечує можливість створення нових класів на основі вже існуючих. При цьому новий клас успадковує характеристики та поведінку базового класу, що дозволяє уникати дублювання коду та формувати ієрархічну структуру програмних компонентів.

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		45

3.2.4 Поліморфізм

Поліморфізм дає змогу використовувати єдиний інтерфейс для роботи з об'єктами різних типів. Це забезпечує гнучкість архітектури програмного забезпечення та спрощує його розширення в майбутньому.

3.2.5 Використання принципів ООП у програмі

Під час розробки програмного комплексу для роботи з графо-логічними моделями було використано основні принципи об'єктно-орієнтованого програмування, зокрема інкапсуляцію та абстракцію. У подальших версіях системи передбачається впровадження механізмів наслідування та поліморфізму для підвищення гнучкості програмної архітектури.

Реалізація інкапсуляції

У програмному комплексі механізм інкапсуляції використовується для забезпечення захисту даних, що описують структуру та параметри GL-моделей.

Інформація про вершини, ребра, реберні функції та інші характеристики моделі зберігається всередині відповідних класів і недоступна для прямого редагування із зовнішнього середовища. Доступ до цих даних здійснюється лише через спеціально визначені методи.

Такий підхід дозволяє підтримувати цілісність моделі та мінімізувати ризик виникнення помилок, пов'язаних із некоректною зміною внутрішнього стану об'єктів.

Реалізація абстракції

Для спрощення взаємодії з графо-логічними моделями в програмному комплексі створено набір класів із чітко визначеними функціональними можливостями.

Кожен клас реалізує окремий аспект роботи системи та надає користувачеві лише необхідний набір операцій. Завдяки цьому складність внутрішньої реалізації приховується, а робота з програмним комплексом стає

більш зрозумілою та зручною.

3.3 Особливості програми

Важливою особливістю розробленого програмного комплексу є не лише можливість автоматизованої побудови та модифікації графо-логічних моделей, але й реалізація механізму перевірки їх коректності. На відміну від багатьох існуючих програмних рішень, які обмежуються лише генерацією структури моделі, створена система дозволяє виконувати комплексний аналіз отриманих результатів та виявляти можливі помилки ще на етапі проектування.

Перевірка коректності є надзвичайно важливим етапом роботи з GL-моделями, оскільки навіть незначні помилки у структурі графа або логічних виразах реберних функцій можуть призвести до отримання некоректних результатів під час оцінювання надійності багатопроцесорної системи. У такому випадку модель перестане адекватно відображати поведінку реальної системи, що може спричинити помилкові висновки щодо її відмовостійкості.

Розроблений програмний комплекс дозволяє автоматично аналізувати побудовану модель та визначати її відповідність заданим параметрам системи. Зокрема, виконується перевірка структури графа, коректності реберних функцій, відповідності моделі заданому типу системи, а також правильності реакції моделі на виникнення відмов окремих компонентів.

Особлива увага приділяється аналізу зв'язності графа для різних векторів станів системи. Програма автоматично генерує необхідні комбінації станів процесорів і перевіряє, чи відповідає поведінка моделі теоретичним вимогам. Для базових систем типу $K(m,n)$ виконується контроль того, що система залишається працездатною при відмові не більше ніж m процесорів та переходить у стан відмови у випадках, визначених її структурою та логікою роботи.

Додатковою перевагою є можливість виявлення некоректно сформованих реберних функцій. Програмний комплекс аналізує їхню поведінку для різних

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		47

наборів вхідних даних та визначає випадки, коли логічний опис ребра не відповідає заданим умовам функціонування системи. Це значно спрощує процес налагодження моделі та зменшує ймовірність появи прихованих помилок.

Реалізований механізм перевірки також дозволяє оцінювати результати модифікації GL-моделей. Після внесення змін до структури графа або реберних функцій програма автоматично виконує повторний аналіз коректності, що дає змогу переконатися у збереженні необхідних властивостей моделі та її відповідності початковим вимогам.

Завдяки інтеграції засобів побудови, модифікації та автоматизованої перевірки коректності розроблений програмний комплекс є не лише інструментом створення GL-моделей, а й повноцінним середовищем їх дослідження. Це підвищує достовірність отриманих результатів, скорочує час розробки моделей та мінімізує ризик використання некоректних моделей під час оцінювання надійності відмовостійких багатопроцесорних систем.

Таким чином, можливість автоматичної перевірки коректності побудованих графо-логічних моделей є однією з ключових переваг розробленого програмного комплексу, яка вирізняє його серед аналогічних рішень та підвищує практичну цінність для дослідження і проєктування відмовостійких багатопроцесорних систем.

3.4 Аналіз отриманих результатів

У результаті розробки програмного комплексу було реалізовано систему побудови, модифікації та перевірки коректності графо-логічних моделей відмовостійких багатопроцесорних систем. Проведене тестування підтвердило працездатність усіх основних функціональних модулів та можливість їх практичного використання для дослідження надійності багатопроцесорних систем.

Під час виконання експериментів було перевірено коректність алгоритмів побудови GL-моделей для базових систем типу $K(m,n)$. Отримані результати

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		48

показали, що програмний комплекс успішно формує графові структури відповідно до заданих параметрів системи та забезпечує правильне формування реберних функцій для кожного елемента моделі.

Окрему увагу було приділено роботі алгоритмів перевірки коректності побудованих моделей. У процесі тестування було встановлено, що система здатна автоматично виявляти помилки у структурі графа, некоректні реберні функції та невідповідність моделі заданим параметрам відмовостійкості. Це значно знижує ймовірність використання помилкових моделей під час подальших досліджень і підвищує достовірність отриманих результатів.

Результати перевірки зв'язності графів показали високу ефективність реалізованих алгоритмів. Для кожного сформованого вектора стану система коректно визначає наявність або відсутність зв'язності графа та робить висновок щодо працездатності відповідної багатопроцесорної системи. Отримані результати повністю узгоджуються з теоретичними положеннями графо-логічного моделювання.

Важливим результатом роботи програмного комплексу стала можливість автоматизованого аналізу поведінки системи в потоці відмов. Програма дозволяє досліджувати вплив відмов окремих процесорів на загальний стан системи та визначати критичні комбінації відмов, які призводять до втрати працездатності. Це дає змогу більш детально дослідити характеристики надійності та відмовостійкості проєктованих систем.

Під час тестування також було досліджено роботу механізмів модифікації GL-моделей. Результати показали, що програмний комплекс коректно виконує внесення змін до структури моделі та реберних функцій, після чого автоматично перевіряє відповідність отриманої моделі встановленим вимогам. Завдяки цьому забезпечується збереження працездатності моделі та правильність її подальшого використання.

Практичні випробування підтвердили, що використання автоматизованих алгоритмів значно скорочує час побудови та аналізу графо-логічних моделей

					ІАЛЦ.467200.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		49

порівняно з ручним виконанням аналогічних операцій. Крім того, усувається значна кількість помилок, які можуть виникати через людський фактор під час проєктування складних багатопроцесорних систем.

Особливою перевагою створеного програмного комплексу є поєднання функцій побудови моделей, їх модифікації та автоматизованої перевірки коректності в межах єдиного програмного середовища. На відміну від більшості існуючих рішень, які орієнтовані лише на створення моделей, розроблена система забезпечує повний цикл роботи з GL-моделями, починаючи від їх генерації та завершуючи аналізом правильності отриманих результатів.

Отримані результати свідчать про ефективність запропонованого підходу до автоматизації побудови графо-логічних моделей та підтверджують можливість використання програмного комплексу для дослідження надійності відмовостійких багатопроцесорних систем різного рівня складності.

Таким чином, результати проведених досліджень демонструють, що розроблений програмний комплекс успішно виконує поставлені завдання, забезпечує коректне формування та аналіз GL-моделей, а також може використовуватися як інструмент для подальших наукових досліджень у галузі моделювання та оцінювання надійності багатопроцесорних систем.

ВИСНОВОК

У процесі виконання роботи було проведено дослідження відмовостійких багатопроцесорних систем, методів оцінювання їх надійності та підходів до моделювання їх функціонування в умовах виникнення відмов. Встановлено, що для розрахунку надійності небазових багатопроцесорних систем існує велика кількість методів, проте одним із найбільш ефективних і достовірних є графо-логічний підхід, який дозволяє гнучко враховувати складні взаємозв'язки елементів системи.

У теоретичній частині роботи розглянуто основні засади побудови відмовостійких багатопроцесорних систем, зокрема їх структуру, способи резервування та механізми реконфігурації. Показано, що базові системи мають добре досліджені та формалізовані алгоритми побудови й аналізу, тоді як небазові системи відзначаються складною поведінкою та взаємодією компонентів, що суттєво ускладнює їх дослідження і потребує використання спеціалізованих методів моделювання.

У роботі виконано аналіз основних підходів до оцінювання надійності, зокрема аналітичних, статистичних і комбінованих. Обґрунтовано, що для систем зі складною структурою доцільно застосовувати моделі, які поєднують структурний аналіз із моделюванням випадкових процесів. Саме таким інструментом виступають графо-логічні моделі, що забезпечують високу адекватність опису складних багатопроцесорних архітектур.

Окрему увагу приділено дослідженню графо-логічних моделей, принципам їх побудови, застосуванню реберних функцій і аналізу зв'язності графа як критерію працездатності системи. Розглянуто як базові моделі з відомими алгоритмами побудови, так і підходи до моделювання небазових систем. Для останніх було детально проаналізовано методи, які дозволяють відтворити їх складну поведінку та забезпечити коректне представлення структури й функціонування.

Практичним результатом дослідження стала розробка програмного комплексу, призначеного для автоматизованої побудови, модифікації та аналізу графо-логічних моделей. Реалізовані алгоритми забезпечують як формування моделей, так і їх перевірку на відповідність заданим параметрам відмовостійкості, що підвищує достовірність отриманих результатів.

Проведені експериментальні дослідження підтвердили правильність функціонування запропонованих підходів і засвідчили, що моделі, розглянуті у роботі, є коректними та адекватно відображають поведінку систем у різних умовах експлуатації. Отримані результати також довели ефективність застосування графо-логічного методу для аналізу надійності як базових, так і небазових багатопроцесорних систем.

Таким чином, поставленої мети роботи досягнуто: досліджено методи оцінювання надійності, проаналізовано особливості базових і небазових систем, обґрунтовано доцільність використання графо-логічного підходу та підтверджено його ефективність як на теоретичному, так і на експериментальному рівнях. Отримані результати можуть слугувати основою для подальшого розвитку методів моделювання та автоматизованого аналізу складних відмовостійких багатопроцесорних систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

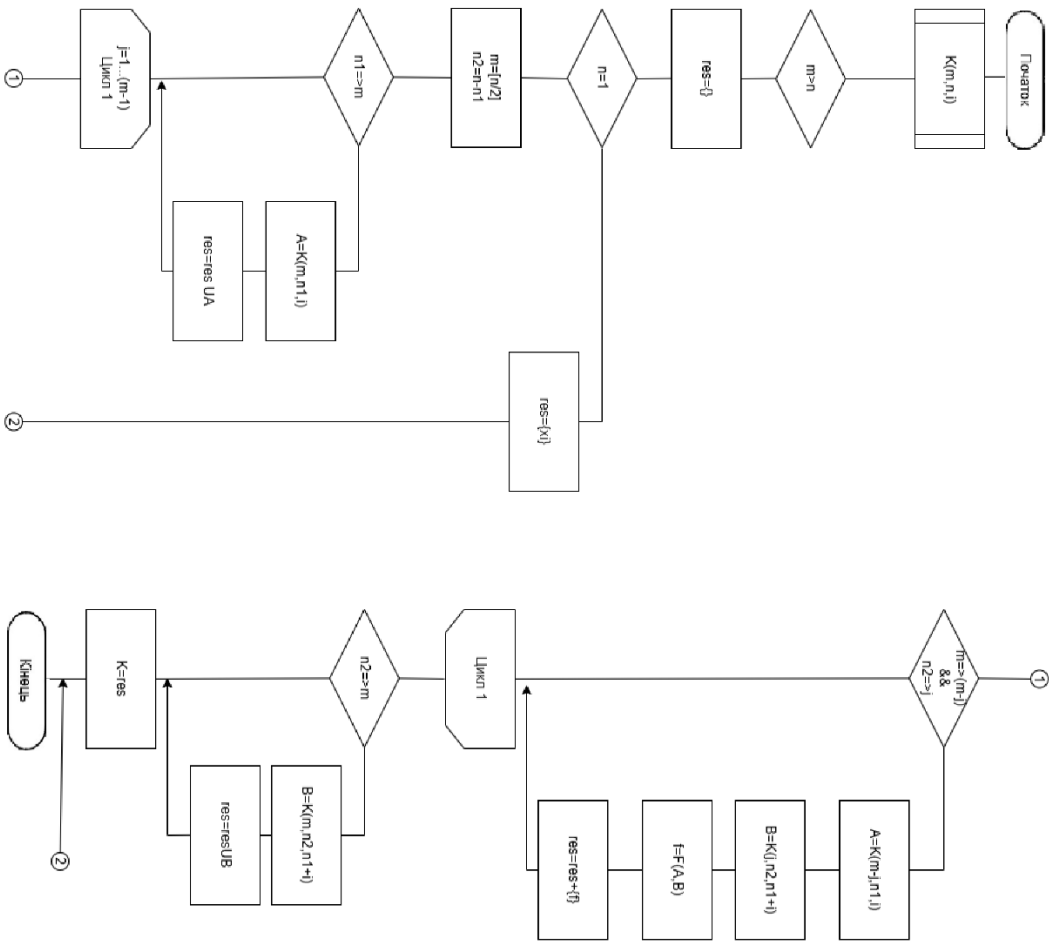
1. Романкевич В. О. Про один спосіб побудови GL-моделей відмовостійких багатопроцесорних систем // Радіоелектронні і комп'ютерні системи. – 2006.
2. Romankevitch A., Morozov K., Romankevich V., Halytskyi D., Eleftherios Z. Improved GL-Model of Behavior of Complex Multiprocessor Systems in Failure Flow. // Advances in Computer Science for Engineering and Education VI (ICCSEEA 2023), LNDECT, vol 181. – 2023.
3. Romankevitch A., Morozov K., Mykytenko S., Kovalenko O., «On the cascade GL-model and its properties» / Applied Aspects of Information Technology. – 2022.
4. Romankevich A.M., Morozov K.V., Romankevich. V.A. Graph-Logic Models of Hierarchical Fault-Tolerant Multiprocessor Systems. IJCSNS International Journal of Computer Science and Network Security. 2019.
5. Lee W.S., Grosh D.L., Tillman F.A., Lie C.H. Fault tree analysis, methods and applications:a review. *IEEE Transactions on Reliability*. 1985.
6. Морозов К., Романкевич А., Романкевич В. Двоїста GL-модель та її застосування для побудови моделей поведінки багатомодульних систем у потоці відмов. – 2023.
7. К. В. Морозов, О. М. Романкевич, В. О. Романкевич Про характер впливу модифікації реберних функцій GL-моделі на її поведінку в потоці відмов // Радіоелектронні і комп'ютерні системи. – 2016.
8. Коваленко, О. П., Романкевич В. О. Алгоритм і програма формування діагностичних графів багатопроцесорних систем : дипломний проект бакалавра : 123 Комп'ютерна інженерія / Коваленко Олена Павлівна. – Київ, 2019.

9. К. В. Морозов, В. О. Романкевич, К. Р. Потапова Про модифікацію графо-логічної моделі // Міжнародний науково-технічний журнал “Інформаційні технології та комп’ютерна інженерія”. – 2015.
10. В. О. Романкевич, А. О. Кононова Про один метод перетворення GL-моделей поведінки відмовостійких багатопроцесорних систем в потоці відмов // Радіоелектронні і комп’ютерні системи. – 2007.
11. В. О. Романкевич, К. В. Морозов, А. П. Фесенюк. Про один метод модифікації реберних функцій GL-моделей // Радіоелектронні і комп’ютерні системи. – 2014.
12. A method for constructing GL-models of behavior under failure flow for complex non-basic fault-tolerant multiprocessor systems [Електронний ресурс]. – Режим доступу:
<https://aait.od.ua/index.php/journal/article/view/226/195> (дата звернення: 30.05.2026).
13. Fault Tree Analysis (FTA) [Електронний ресурс]. – Режим доступу:
<https://www.creativesafetysupply.com/glossary/fault-tree-analysis-fta/> (дата звернення: 05.06.2026).
14. On the method of building of non-basic GL-models which are formed on combination of edge functions of basic models [Електронний ресурс]. – Режим доступу: <https://aait.od.ua/index.php/journal/article/view/151/151> (дата звернення: 05.06.2026).
15. Understanding Reliability Block Diagrams [Електронний ресурс]. – Режим доступу: <https://accendoreliability.com/understanding-reliability-block-diagrams/> (дата звернення: 05.06.2026).

ДОДАТКИ

Додаток 1.

Копії графічного матеріалу



Ім'я Морені	Підписав/створив	Ім'я Ін. №	Ім'я Науки	Підписав/створив
-------------	------------------	------------	------------	------------------

IA.UPL. 467200.005 Д1									
Побудова базових									
модулів.									
Схема виконання									
№	Питання	Відповідь	№	Питання	Відповідь	№	Питання	Відповідь	№
1	Питання 1.1	Відповідь 1.1	2	Питання 1.2	Відповідь 1.2	3	Питання 1.3	Відповідь 1.3	4
5	Питання 1.5	Відповідь 1.5	6	Питання 1.6	Відповідь 1.6	7	Питання 1.7	Відповідь 1.7	8
9	Питання 1.9	Відповідь 1.9	10	Питання 1.10	Відповідь 1.10	11	Питання 1.11	Відповідь 1.11	12
13	Питання 1.13	Відповідь 1.13	14	Питання 1.14	Відповідь 1.14	15	Питання 1.15	Відповідь 1.15	16
17	Питання 1.17	Відповідь 1.17	18	Питання 1.18	Відповідь 1.18	19	Питання 1.19	Відповідь 1.19	20
21	Питання 1.21	Відповідь 1.21	22	Питання 1.22	Відповідь 1.22	23	Питання 1.23	Відповідь 1.23	24
25	Питання 1.25	Відповідь 1.25	26	Питання 1.26	Відповідь 1.26	27	Питання 1.27	Відповідь 1.27	28
29	Питання 1.29	Відповідь 1.29	30	Питання 1.30	Відповідь 1.30	31	Питання 1.31	Відповідь 1.31	32
33	Питання 1.33	Відповідь 1.33	34	Питання 1.34	Відповідь 1.34	35	Питання 1.35	Відповідь 1.35	36
37	Питання 1.37	Відповідь 1.37	38	Питання 1.38	Відповідь 1.38	39	Питання 1.39	Відповідь 1.39	40
41	Питання 1.41	Відповідь 1.41	42	Питання 1.42	Відповідь 1.42	43	Питання 1.43	Відповідь 1.43	44
45	Питання 1.45	Відповідь 1.45	46	Питання 1.46	Відповідь 1.46	47	Питання 1.47	Відповідь 1.47	48
49	Питання 1.49	Відповідь 1.49	50	Питання 1.50	Відповідь 1.50	51	Питання 1.51	Відповідь 1.51	52
53	Питання 1.53	Відповідь 1.53	54	Питання 1.54	Відповідь 1.54	55	Питання 1.55	Відповідь 1.55	56
57	Питання 1.57	Відповідь 1.57	58	Питання 1.58	Відповідь 1.58	59	Питання 1.59	Відповідь 1.59	60
61	Питання 1.61	Відповідь 1.61	62	Питання 1.62	Відповідь 1.62	63	Питання 1.63	Відповідь 1.63	64
65	Питання 1.65	Відповідь 1.65	66	Питання 1.66	Відповідь 1.66	67	Питання 1.67	Відповідь 1.67	68
69	Питання 1.69	Відповідь 1.69	70	Питання 1.70	Відповідь 1.70	71	Питання 1.71	Відповідь 1.71	72
73	Питання 1.73	Відповідь 1.73	74	Питання 1.74	Відповідь 1.74	75	Питання 1.75	Відповідь 1.75	76
77	Питання 1.77	Відповідь 1.77	78	Питання 1.78	Відповідь 1.78	79	Питання 1.79	Відповідь 1.79	80
81	Питання 1.81	Відповідь 1.81	82	Питання 1.82	Відповідь 1.82	83	Питання 1.83	Відповідь 1.83	84
85	Питання 1.85	Відповідь 1.85	86	Питання 1.86	Відповідь 1.86	87	Питання 1.87	Відповідь 1.87	88
89	Питання 1.89	Відповідь 1.89	90	Питання 1.90	Відповідь 1.90	91	Питання 1.91	Відповідь 1.91	92
93	Питання 1.93	Відповідь 1.93	94	Питання 1.94	Відповідь 1.94	95	Питання 1.95	Відповідь 1.95	96
97	Питання 1.97	Відповідь 1.97	98	Питання 1.98	Відповідь 1.98	99	Питання 1.99	Відповідь 1.99	100

[illegible]

Додаток 2.

Презентація

Програма побудови та перевірки GL-моделей складних небазових відмовостійких багатопроцесорних систем

Виконав Студент 4 курсу КВ-22
Кошулько Владислав

Керівник: Доцент каф. СПІСКС, к.т.н.,
Морозов К.В.

2026

Актуальність теми

- Вибрана тема є потрібною та необхідною для подальшого розвитку в цій галузі. Адже з стрімким розвитком технологій програма яка буде ефективно та коректно будувати моделі складних небазових для перевірки їх надійності

Мета роботи

- ▶ Метою є розробка програми для побудови GL-моделей складних небазових відмовостійких систем та перевірки побудованих моделей на коректність

Технічні вимоги

- ▶ Для працездатності програми потрібні:
- ▶ Процес на 4 ядра або більше
- ▶ 8ГБ оперативної пам'яті або більше
- ▶ Операційна система Windos, macOS, Linux

Вибір мови програмування

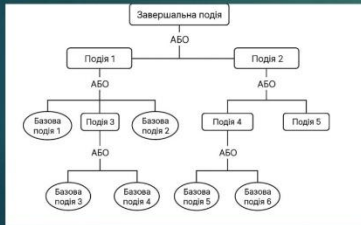
- Для написання самої програми доцільним було обрати мову програмування Python. Адже вона є високорівневою мовою програмування загального призначення, яка здобула широку популярність завдяки простоті використання, зрозумілому синтаксису та значним можливостям для розв'язання різноманітних прикладних і наукових задач. Завдяки своїм особливостям ця мова активно застосовується у сфері наукових досліджень, автоматизації процесів, аналізу даних, машинного навчання та розробки складних програмних систем.

ВБС

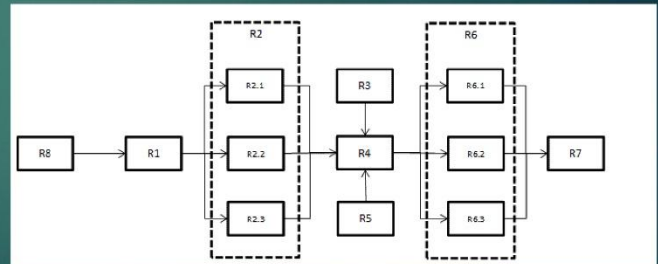
- Всі методи що розглянуто в роботі використовуються для побудови моделей ВБС- вбдмовостійких багатопроцесорних систем. Вони зараз використовуються у багатьох важливих та небезпечних сферах таких як космічна промисловість, виробництво дронів і інших де потрібна безвідмовна робота.

Існуючі рішення та їх аналіз

- Для побудови моделей складних небазових систем вже існує достатньо методів які було розглянуто в цій роботі



Приклад діаграми дерева відмови



Приклад методу блок-схеми

Недоліки розглянутих методів

- Головними недоліками інших розглянутих методів можна навзати складність реалізації, менша гнучкість, порівнюючи з вибраним методом менша ефективність.

Обраний метод

- У цій роботі розглянуто метод побудови GL-моделей для оцінки надійності системи. Він є більш ефективним у порівнянні з іншими. Цей метод пропонує будувати базові моделі заданої системи та об'єднувати їх, також за необхідності модифікувати щоб досягнути максимальної точності

Недоліки обраного методу

Попри значні переваги, графо-логічні моделі мають і певні недоліки. До основних обмежень належать:

- висока обчислювальна складність під час моделювання великих систем;
- необхідність ретельного проєктування та налаштування реберних функцій;
- підвищена чутливість до похибок у вихідних параметрах моделі;
- збільшення складності аналізу зі зростанням кількості елементів системи.

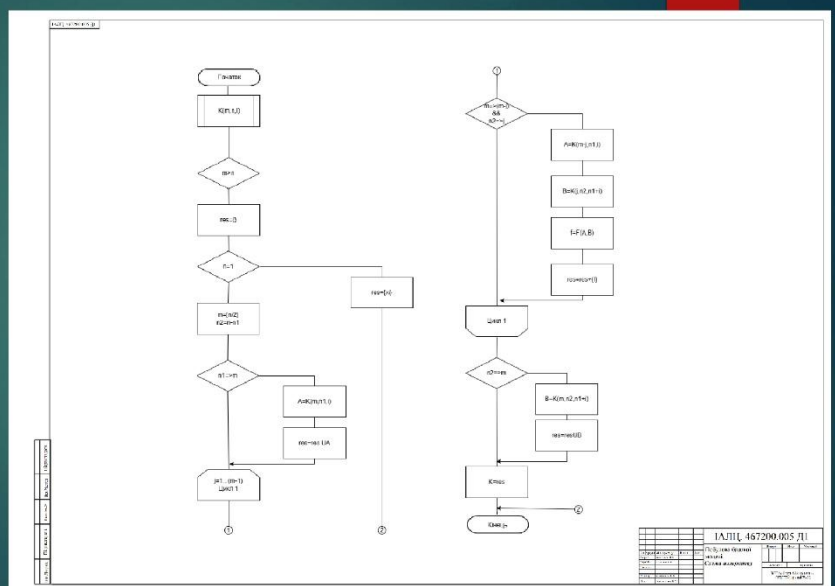
Алгоритми використані у програмі

Програма використовує алгоритми:

- ▶ для побудови базових моделей,
- ▶ для побудови небазових моделей,
- ▶ модифікація небазових моделей,
- ▶ Перевірка моделі на коректність

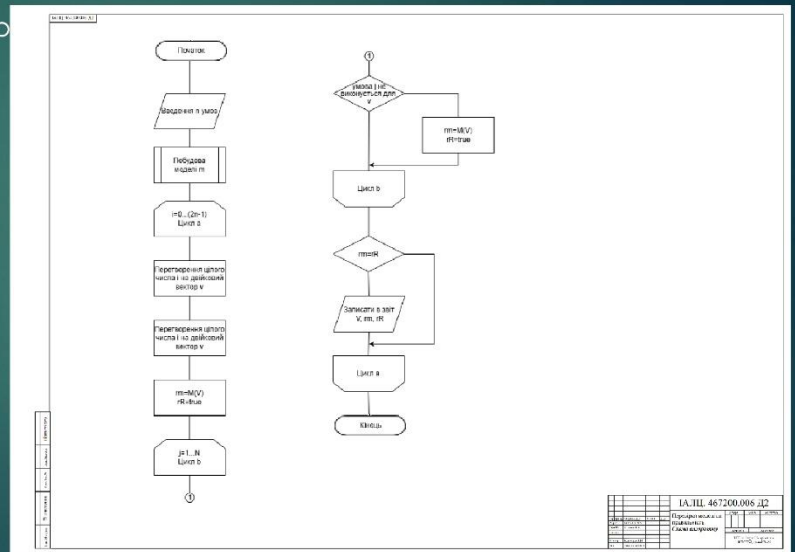
Побудова базової моделі

На даному кресленні відображено побудову базової моделі яка буде використовуватись для побудови небазової моделі



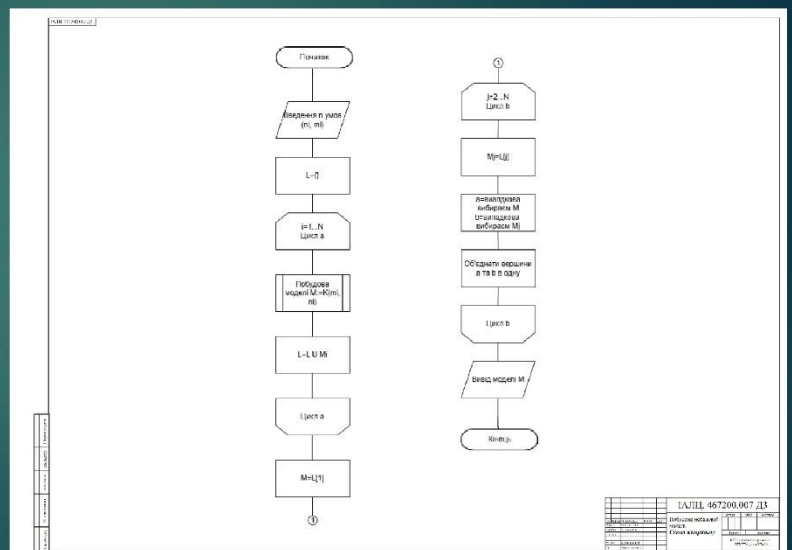
Перевірка моделі на правильність

На даному креслені відображено алгоритм який перевіряє GL-модель на правильність побудови



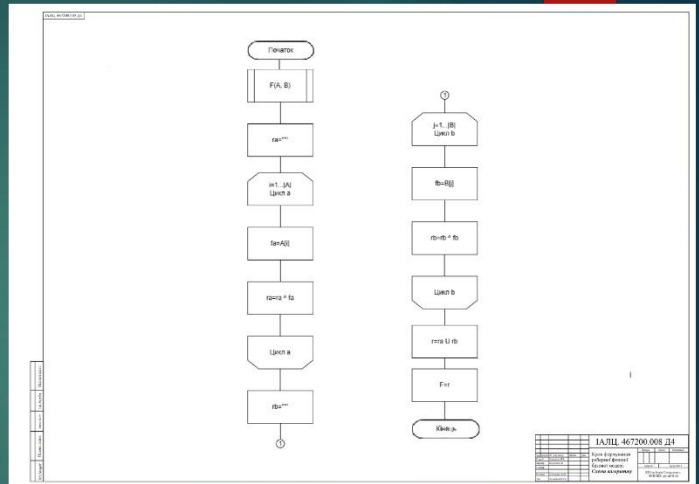
Побудова небазової моделі

На даному креслені відображено алгоритм який будує небазову GL-модель для подальшої роботи



Крок формування реберної функції базової моделі

На цьому кресленні відображено алгоритм побудови реберної функції базової моделі



Сфери застосування програми

- ▶ Системи керування енергетичними об'єктами
- ▶ Медичні діагностичні комплекси
- ▶ Центри обробки даних
- ▶ Космічна промисловість

Можливості програми

- Програма здатна ефективно будувати коректні GL-моделі заданих складних небазових систем за допомогою розглянутих в роботі алгоритмів

Висновок

- Під час виконання роботи були проведені аналізи ВБС та сфери де вони застосовуються. Було розглянуто методи моделювання даних систем з метою перевірити як вони поведуться в потоці відмов, тобто перевірити їхню надійність. В роботі для детальнішого аналізу обрано метод побудови графо-логічних моделей який є більш ефективним у порівнянні з іншими. Також в роботі розглянуто алгоритми якими користується програма для виконня поставлених задач.