

Міністерство освіти і науки України

Український державний університет науки і технологій

Комп'ютерні технології і системи

Електронні обчислюванні системи

Пояснювальна записка

до кваліфікаційної роботи

Магістр

До захисту
23.01.2021

На тему: Розробка моделей складних інформаційно-комунікаційних систем та їх дослідження для використання у навчальному процесі

за освітньою програмою Комп'ютерна інженерія

зі спеціальності: 123 Комп'ютерна інженерія

Виконав: студент КС 2221 групи:

Ску

Сергій Коцюбинський

Керівник:

[Signature]

професор Анатолій Косолапов

Нормоконтролер:

[Signature]

доцент Володимир Шаповалов

Консультанти:

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

Студент

Ску

Дніпро – 2024 рік

Ministry of Education and Science of Ukraine
Ukrainian State University of Science and Technologies

Computer technologies and systems

(faculty)

Electronic computing systems

(department)

Explanatory Note
to Master's Bachelor's Thesis
Master's degree
(higher education degree)

on the topic: Development of models of complex information and communication systems and their research for use in the educational process

according to educational curriculum Computer engineering
in the Speciality: 123 Computer engineering
(speciality and its code)

Done by the student CS 2221 of the group:

Serhii Kotsiubynskyi

Scientific Supervisor::

Anatolii Kosolapov

(position, name, surname)

Normative controller:

docent Volodymyr Shapovalov

Supervisors Consultants:

(position, name, surname)

(Chapter title heading)

/ /

(position, name, surname)

(Chapter title heading)

/ /

(position, name, surname)

(Chapter title heading)

/ /

(position, name, surname)

(Chapter title heading)

/ /

(position, name, surname)

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет: Комп'ютерні технології і системи

Кафедра: Електронні обчислювальні машини

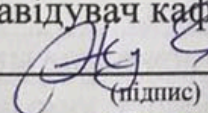
Рівень вищої освіти: Другий (магістерський)

Освітня програма: Комп'ютерна інженерія

Спеціальність: 123 Комп'ютерна інженерія

ЗАТВЕРДЖУЮ

Завідувач кафедри ЕОМ


(підпис)

Ігор Жуковицький
(Ім'я ПРІЗВИЩЕ)

Дата 15.11.2023

ЗАВДАННЯ

на кваліфікаційну роботу

Магістр

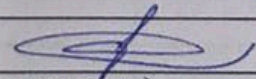
(ступінь вищої освіти)

студенту Коцюбинському Сергію Сергійовичу
(Прізвище, Ім'я Побатькові)

1. Тема роботи: Розробка моделей складних інформаційно-комунікаційних систем та їх дослідження для використання у навчальному процесі

Керівник роботи: Косолапов А. А., д.т.н., професор

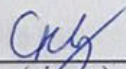
(Прізвище, Ім'я, Побатькові, науковий ступінь, вчене звання)



КАЛЕНДАРНИЙ ПЛАН

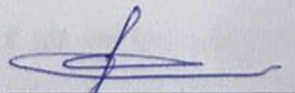
№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вступ		
2	Аналіз і вибір програмних засобів моделювання інформаційно-комунікаційних систем		
3	Дослідження основних механізмів функціонування складних інформаційних систем	27.11.2023	30%
4	Аналіз функціонування інформаційно-комунікаційної системи на прикладі веб-серверу.	18.12.2023	60%
5	Рекомендації та висновки		
6	Підготовка презентації та доповіді	08.01.2024	100%
7	Подання кваліфікаційної роботи до кафедри	08.01.2024	
8	Захист кваліфікаційної роботи на засіданні Екзаменаційної комісії		

Студент


(підпис)

Сергій КОЦЮБИНСЬКИЙ
(Ім'я ПРІЗВИЩЕ)

Керівник роботи


(підпис)

Анатолій КОСОЛАПОВ
(Ім'я ПРІЗВИЩЕ)

56 ст., 22 рис., 5 табл., 20 джерел.

Об'єкт дослідження - інформаційно-комунікаційні системи.

Мета даної дипломної роботи є аналіз, вибір та використання програмних засобів моделювання, вивчення особливостей інструментів для дослідження механізмів функціонування інформаційних систем, побудова імітаційних моделей та аналіз результатів моделювання.

Цілі дипломної роботи:

- Аналіз програмних засобів моделювання та вибір оптимального інструменту.
- Побудова імітаційної моделі інформаційно-комунікаційної системи.
- Дослідження та аналіз функціонування системи з використанням побудованої моделі.

Методи дослідження: Системний аналіз розвитку основ моделювання інформаційно-комунікаційних систем.

Виконано аналіз основних засобів моделювання інформаційно-комунікаційних систем серед яких перевагу віддано пакету Java Modeling Tools. Побудова імітаційних моделей розглядається на прикладі двох серверної ІКС, що працює з дисковим простором за схемою RAID0 та описаним механізмом розподілу заявок.

Результати роботи можуть бути використані в процесі концептуального проектування інформаційно-керуючих систем для підприємств промисловості та транспорту

Ключові слова: ІМІТАЦІЙНА МОДЕЛЬ, JAVA MODELING TOOLS, ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ СИСТЕМИ, АНАЛІЗ СИСТЕМИ, ДОСЛІДЖЕННЯ МЕХАНІЗМІВ.

ЗМІСТ

ВСТУП.....	8
1 АНАЛІЗ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ.....	10
1.1 Характеристика моделювання інформаційно-комунікаційних систем...	10
1.1.1 Класифікація ІКС	10
1.1.2 Основні відмінності та спільні риси моделей	12
1.2 Аналіз та порівняння програмних засобів моделювання ІКС	16
1.2.1 AnyLogic	16
1.2.2 MATLAB (Simulink)	18
1.2.3 OPNET	19
1.2.4 Java Modeling Tools (JMT)	20
1.3 Вибір і характеристика Java Modeling Tools (JMT) як інструменту для дослідження	22
2 ДОСЛІДЖЕННЯ ОСНОВНИХ МЕХАНІЗМІВ ФУНКЦІОНУВАННЯ СКЛАДНИХ ІНФОРМАЦІЙНИХ СИСТЕМ.....	25
2.1 Моделювання системи в середовищі JMT на прикладі розробки моделі веб- серверу	25
2.1.1 Опис процесу створення імітаційної моделі	25
2.1.2 Таблиці з характеристиками елементів системи.	29
3 АНАЛІЗ ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНОЇ СИСТЕМИ НА ПРИКЛАДІ ВЕБ-СЕРВЕРУ.....	40
3.1 Опис результатів моделювання	40
3.2 Аналіз та інтерпретація результатів дослідження	52
ВИСНОВКИ.....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	54

ВСТУП

Сучасний світ неможливо уявити без інформаційно-комунікаційних систем, які є необхідною складовою сучасного суспільства та бізнес-середовища. Інформаційні технології, мережі зв'язку, комп'ютерні системи та програмне забезпечення визначають спосіб, яким ми спілкуємося, працюємо та вивчаємо нові знання. Враховуючи постійний розвиток цих систем та їх вплив на різні аспекти суспільства, дослідження та розробка моделей інформаційно-комунікаційних систем стають важливою складовою сучасної науки та технологій.

Об'єкт дослідження - інформаційно-комунікаційні системи.

Предмет дослідження - моделі основних механізмів функціонування інформаційно-комунікаційних систем та їх дослідженню з метою використання у навчальному процесі.

Мета даної дипломної роботи - є аналіз, вибір та використання програмних засобів моделювання, вивчення особливостей інструментів для дослідження механізмів функціонування інформаційних систем, побудова імітаційних моделей та аналіз результатів моделювання.

Цілі дипломної роботи:

- Аналіз програмних засобів моделювання та вибір оптимального інструменту.
- Побудова імітаційної моделі інформаційно-комунікаційної системи.
- Дослідження та аналіз функціонування системи з використанням побудованої моделі.

Методи дослідження: Системний аналіз розвитку основ моделювання інформаційно-комунікаційних систем.

Завдання:

1. Порівняльний аналіз програмних засобів для моделювання і вибір раціонального інструменту на основі критеріїв ефективності та доступності.
2. Розробка імітаційної моделі інформаційно-комунікаційної системи з використанням обраного програмного засобу.
3. Проведення імітаційних експериментів та збір даних для подальшого аналізу.
4. Аналіз отриманих результатів моделювання.

Робота складається з вступу, трьох розділів, висновків і списку використаної літератури (20).

В роботі три розділи:

1. В першому розділі виконується аналіз і вибір програмних засобів моделювання інформаційно-комунікаційних систем. В якості інструментального середовища обрано пакет Java Modeling Tools (JMT).
2. Другий розділ присвячено дослідженню основних механізмів функціонування складних інформаційних систем.
3. В третьому розділі проведено аналіз функціонування інформаційно-комунікаційної системи на прикладі веб-серверу.
4. Завершається робота загальними висновками і списком використаних джерел.

1 АНАЛІЗ І ВИБІР ПРОГРАМНИХ ЗАСОБІВ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ

1.1 Характеристика моделювання інформаційно-комунікаційних систем

1.1.1 Класифікація ІКС

Класифікація інформаційно-комунікаційних систем є основою для розуміння їхньої природи, функціональності та ролі в сучасному суспільстві.

Класифікація за функціональним призначенням:

- *Інформаційні системи* - ці системи займаються збором, обробкою, збереженням та передачею інформації. Прикладами є бази даних, системи управління документами та інші.

- *Комунікаційні системи* - основною метою цих систем є передача інформації між вузлами чи користувачами. Сюди входять телефонні мережі, Інтернет, електронна пошта та інші засоби зв'язку.

- *Інформаційно-комунікаційні системи (ІКС)* - ця категорія об'єднує системи, які поєднують у собі функціональність інформаційних та комунікаційних систем. Такі системи часто використовуються в бізнесі для підтримки різних процесів.

Класифікація за обсягом обробки інформації:

- *Малий обсяг інформації* - ці системи призначені для обробки невеликого обсягу даних та можуть використовуватися на персональних комп'ютерах чи в невеликих офісах.

- *Середній обсяг інформації* - ця категорія включає системи, які спрямовані на обробку середнього обсягу даних та можуть використовуватися у великих підприємствах та організаціях.

- *Великий обсяг інформації* - системи цієї категорії призначені для обробки великого обсягу даних, таких як обчислення на великих кластерах серверів чи системи обробки Big Data [3].

Класифікація за типом даних [7]:

- *Текстові системи* - спеціалізовані на обробці та зберіганні текстової інформації, такі як документи, повідомлення та статті.

- *Графічні системи* - орієнтовані на створення, зберігання та обробку графічних об'єктів, зображень та відео.

- *Числові системи* - призначені для обробки числових даних, включаючи обчислення та аналіз числових параметрів.

Ця класифікація допомагає розуміти, які типи інформації можуть бути оброблені та збережені в рамках конкретних інформаційно-комунікаційних систем.

Класифікація за масштабом та діапазоном [7]:

- *Локальні системи* - обмежені в масштабі і призначені для використання в межах окремого місця або офісу.
- *Регіональні системи* - призначені для використання в певному регіоні чи області, охоплюють декілька місць.
- *Глобальні системи* - мають глобальний охоплення і забезпечують зв'язок та обробку даних на всій планеті.

Класифікація інформаційно-комунікаційних систем становить фундаментальну базу для подальшого аналізу та дослідження інформаційно-комунікаційних систем у даній дипломній роботі [8].

1.1.2 Основні відмінності та спільні риси моделей

Моделі інформаційно-комунікаційних систем можуть значно відрізнятися за різними параметрами.

Першою важливою відмінністю є рівень абстракції. Деякі моделі можуть бути високо абстрактними, надаючи загальний огляд інформаційно-комунікаційної системи (ІКС). У таких моделях деталізація технічних аспектів обмежені. З

іншого боку, деякі моделі можуть бути деталізованими, включаючи конкретні технічні деталі та особливості функціонування ІКС [6].

Другою відмінністю є масштаб моделі. Моделі можуть бути великими, охоплюючи всю інформаційно-комунікаційну систему в цілому. Ці моделі створені для вивчення системи як єдності. У той же час існують моделі, які фокусуються на окремих аспектах або підсистемах ІКС, що може дозволяти докладно аналізувати конкретні частини системи [6].

Третьою відмінністю є часовий горизонт моделі, які спрямовані на аналіз системи в реальному часі, дозволяючи вивчати її функціонування та взаємодію зі справжнім часом. Інші моделі можуть використовувати різний часовий період для вивчення ІКС в різних умовах та в часовому контексті.

Четвертою відмінністю є метод моделювання. Моделі використовують різні методи та підходи до моделювання. Деякі з них застосовують дискретно-подієве моделювання для імітації окремих подій та процесів, інші можуть використовувати системи масового обслуговування для аналізу обробки запитів, або агентне моделювання для моделювання взаємодії окремих агентів.

П'ятою відмінністю є цільовий аспект моделі (моделі можуть бути спрямовані на аналіз продуктивності інформаційно-комунікаційних систем та оцінку їхньої продуктивності). Інші зосереджуються на вивченні надійності, безпеки або інших аспектів функціонування ІКС [6].

Усі ці відмінності впливають на те, яким чином моделі дозволяють аналізувати інформаційно-комунікаційні системи та визначати їхню придатність для конкретних досліджень і завдань.

Розглянемо можливості найбільш поширених з них та виберемо найбільш відповідний програмний засіб для подальшого моделювання ІКС (інформаційно-комунікаційної системи).

Вибір ефективного середовища моделювання вимагає виконання їхньої оцінки за запропонованими критеріями. Для оцінки середовищ моделювання доцільним є прийняття дворівневої градації характеристик "(- | +)".

Нижча оцінка "-" відображає недолік, що виражається у високій складності, неможливості або відсутності підтримки реалізації необхідної якості при моделюванні. Найвища оцінка "+" визначає перевагу середовища над іншими.

Розглянемо можливості програмних засобів, які були зведені у таблиці 1.1.

Таблиця 1.1 - Програмні засоби для моделювання ІКС

Критерій	Програмний засіб			
	OPNET	JMT	MATLAB (Simulink)	AnyLogic
Загальна характеристика програмного моделювання комп'ютерних ІКС				
Фірма розробки	Riverbed Technology	Політехнічний університет Мілана та Імперський коледж Лондона	MathWorks	XJ Technologies
Рік	1986	2009	1984	2000
Мова розробки пакету	C	Java	C	Java
Моделі: <ul style="list-style-type: none"> ● <i>Дискретні моделі</i> - які представляють системи з дискретними подіями, де зміни стану відбуваються в окремі моменти часу. ● <i>Аналітичні моделі</i> - що використовують математичні формули для опису систем. ● <i>Візуальні моделі</i> - які забезпечують графічне представлення систем. 				

<ul style="list-style-type: none"> ● <i>Функціональні моделі</i> - що описують взаємозв'язки між частинами системи. ● <i>Статистичні моделі</i> - для аналізу даних і прийняття рішень на основі статистичної інформації. ● <i>Динамічні моделі</i> - які імітують поведінку системи в часі. ● <i>Структурні моделі</i> - що відображають організацію та взаємодії між компонентами системи. 				
дискретні	+	+	-	+
аналітичні	-	+	-	-
візуальні	-	+	+	+
функціональні	-	+	+	+
статистичні	-	+	-	+
динамічні	-	+	-	+
структурні	-	+	+	-
Складність трудомісткість розробки моделі:				
● візуальний конструктор	-	+	+	+
● програмування	+	+	-	+
● верифікація	-	+	-	+
● відкритість	-	+	+	+
● ресурсоемність	-	-	-	-

Продовження таблиці 1.1

Гнучкість розробки: об'єктний	+	+	+	+
● процесний	-	+	+	+
● агентний	-	+	-	+
Масштабування моделі:				
● ручне	-	+	+	+
● програмне	+	+	-	+
Візуалізація функціонування	-	+	-	+
Моделювання процесів в системі	-	+	+	+
Можливість доробки	-	+	-	-

--	--	--	--	--

Незважаючи на велику різницю між моделями(див. табл. 1.1), існують деякі спільні риси, які характерні для більшості моделей інформаційно-комунікаційних систем:

1. *Математична основа* (більшість моделей базуються на математичних принципах та алгоритмах, що дозволяє використовувати їх для аналізу та прогнозу роботи системи).
2. *Симуляція* (багато моделей використовують симуляційні методи, щоб імітувати функціонування ІКС та вивчити їхню поведінку в різних умовах).
3. *Варіабельність параметрів* (моделі можуть включати варіабельні параметри, які дозволяють аналізувати вплив зміни параметрів на роботу системи).
4. *Аналіз результатів* (однією з головних мет роботи з моделями є аналіз отриманих результатів, щоб зрозуміти, як система працює та як можна покращити її функціонування).

Ця спільна риса визначає методологічний підхід до моделювання ІКС та його роль у вивченні цих систем [7].

У підсумку видно (див.табл 1.1), що перевага в можливостях більш коректного моделювання ІКС на стороні програмного засобу ЛМТ. Найбільш виразною ознакою є можливість доробки та складність і трудомісткість моделі, але для більш детального аналізу познайомимося зі всіма програмними засобами окремо.

1.2 Аналіз та порівняння програмних засобів моделювання ІКС

1.2.1 AnyLogic

AnyLogic - це інструмент для моделювання, який вирізняється своєю спроможністю використовувати багатоагентний підхід у процесі моделювання. Це означає, що він дозволяє моделювати процеси на рівні системи в цілому, та й взаємодію окремих агентів в мережі або системі. Багатоагентний підхід особливо

корисний для дослідження та моделювання складних соціальних, економічних або транспортних систем, де важлива взаємодія окремих суб'єктів [2].

Розглянемо приклад моделі у програмі. Приведена модель "системи черги в поштових відділеннях" (рисунок 1.1).

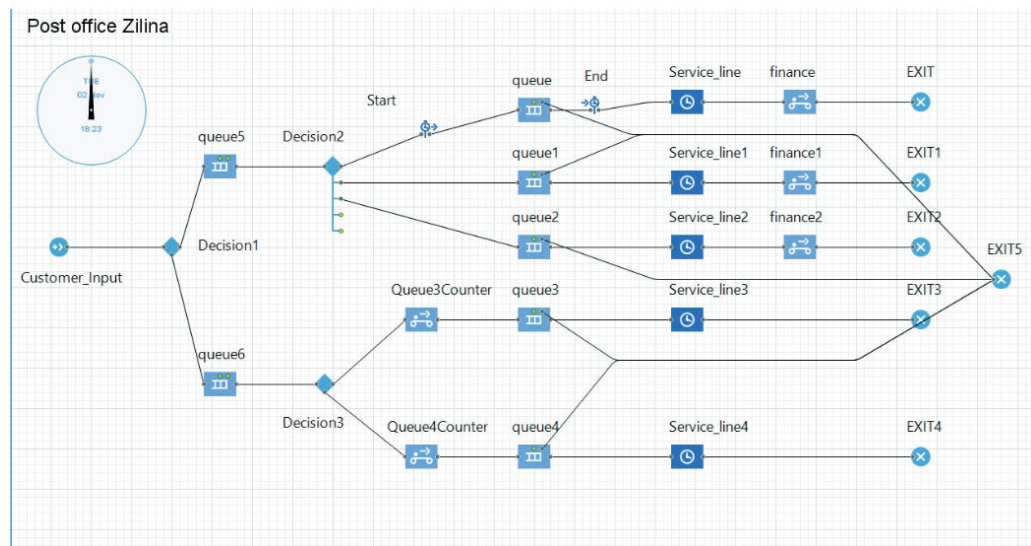


Рисунок 1.1 - Приклад моделі системи черги в поштових відділеннях в Anylogic[18, ст.18]

Однією з переваг AnyLogic є можливість об'єднувати різні типи моделей, включаючи дискретно-подійні, системи масового обслуговування та системи з диференціальними рівняннями. Це дозволяє дослідникам підходити до моделювання з різних точок зору та використовувати підход, який найкраще відповідає конкретній задачі. Крім того, AnyLogic надає інструменти для візуалізації результатів моделювання у вигляді графіків, діаграм, теплових карт та інших візуальних засобів. Це сприяє кращому розумінню результатів та дозволяє зробити зрозумілі висновки з дослідження [17]. Можливість інтеграції AnyLogic з іншими інструментами та мовами програмування, такими як Java, Python, тощо, дозволяє розширювати функціональність інструменту та застосовувати його в різних областях, включаючи наукові дослідження, промислові додатки та бізнес-аналітику [17].

Недоліками AnyLogic є вищі вимоги до ресурсів комп'ютера під час моделювання складних систем, а також більша складність налаштування порівняно з іншими інструментами. Однак з правильними навичками та ресурсами цей інструмент стає потужним засобом для дослідження та моделювання різних видів систем [5].

1.2.2 MATLAB (Simulink)

Simulink - інструмент розроблений компанією MathWorks який відкриває широкі можливості для дослідження та розробки систем з різними рівнями складності.

Однією з ключових переваг Simulink є його інтеграція з MATLAB, що дозволяє користувачам використовувати мову програмування MATLAB для створення складних математичних моделей та обробки даних. Ця інтеграція робить Simulink потужним інструментом для моделювання та аналізу систем з високим рівнем математичної складності. Він також надає багато готових бібліотек та блоків для моделювання різних типів систем, включаючи електричні, механічні, термічні, аеродинамічні та інші, що дозволяє швидко створювати та аналізувати моделі для різних областей інженерії та науки. Окремо слід відзначити можливість візуалізації результатів моделювання в Simulink - інструмент надає різноманітні графічні засоби для відображення даних та аналізу результатів, що дозволяє краще розуміти поведінку системи та приймати інформовані рішення [20].

Розглянемо просту імітаційну модель кулі та балки у MATLAB(рисунк 1.2).

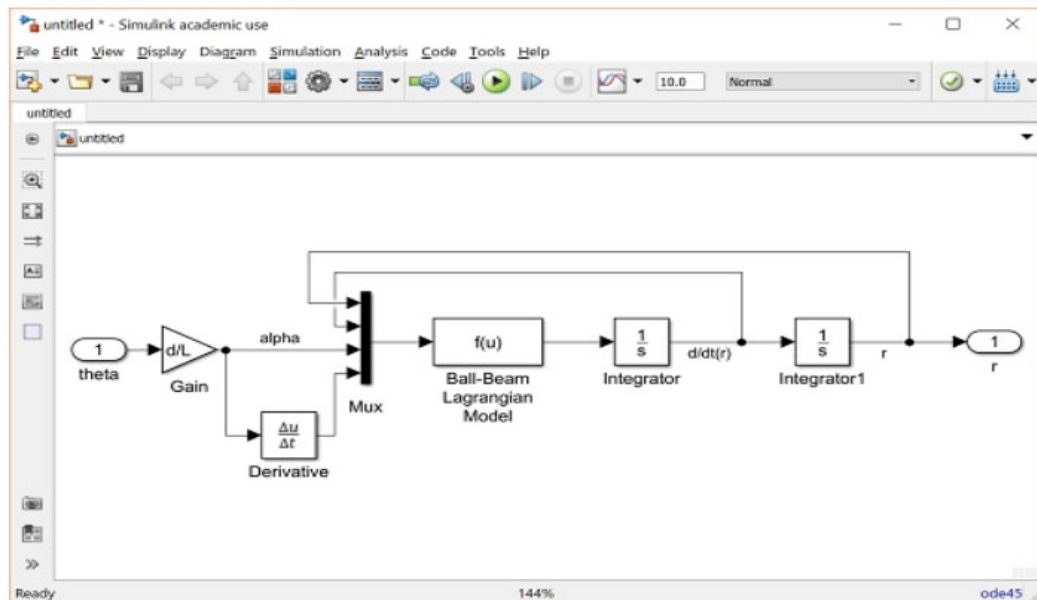


Рис 1.2 - Модель кулі та балки у MATLAB(SIMULINK) [19 ст. 11]

Недоліками Simulink є вищі вимоги до ресурсів комп'ютера під час моделювання складних систем та потреба в спеціалізованих навичках для ефективного використання інструменту. Однак з правильним навчанням та досвідом Simulink стає потужним засобом для дослідження та розробки різних видів систем, що знаходять своє застосування в інженерії, автоматизації, техніці та інших сферах.

1.2.3 OPNET

OPNET (Optimized Network Engineering Tool) - це спеціалізована система моделювання мереж телекомунікацій та аналізу їхньої продуктивності. Цей інструмент розроблений для вивчення різних аспектів мережевого зв'язку, включаючи пропускну здатність, затримку, якість обслуговування, а також оптимізацію ресурсів мережі. OPNET дозволяє інженерам та дослідникам моделювати та аналізувати різні аспекти мережевого зв'язку від простих локальних мереж до великих глобальних телекомунікаційних інфраструктур [20].

Приклад імітаційної моделі мережі університету представлено на рисунку 1.3.

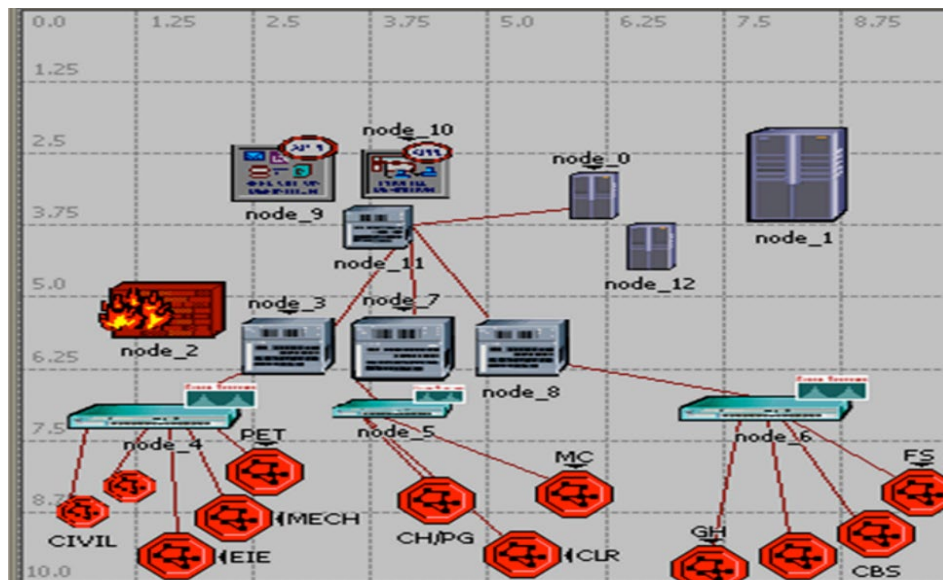


Рисунок 1.3 – Імітаційна модель мережі університету [16, ст. 3]

Перевагою OPNET є його можливість створювати деталізовані моделі мереж, які відображають реальні умови та параметри роботи мережі. Це дозволяє дослідникам виконувати аналіз впливу різних факторів на продуктивність мережі та розробляти стратегії її оптимізації. Він також надає інструменти для візуалізації результатів моделювання, що спрощує розуміння та аналіз даних. Графічний інтерфейс інструменту дозволяє користувачам відобразити результати у вигляді графіків, діаграм, гістограм тощо, що полегшує процес аналізу та прийняття рішень [4].

Недоліками OPNET можуть бути вищі вимоги до обладнання комп'ютера та великий обсяг ресурсів, необхідний для запуску складних моделей мереж. Крім того, використання OPNET вимагає спеціалізованих знань та навичок у галузі телекомунікацій та мережевого інженерінгу [20].

При порівнянні цих систем можна винести специфічні вимоги та завдання дослідження інформаційно-комунікаційних систем. JMT буде корисним для моделювання та аналізу продуктивності мереж, тоді як AnyLogic буде кращим вибором для вивчення взаємодії користувачів в системі [5].

1.2.4 Java Modeling Tools (JMT)

Серед різноманітних систем моделювання інформаційно-комунікаційних систем (ІКС) особливе значення має Java Modelling Tools (JMT). Цей інструмент відзначається своєю важливістю та актуальністю, оскільки він є вільним та відкритим програмним продуктом, який створений спеціально для моделювання та аналізу ІКС. JMT надає користувачам унікальну можливість досліджувати та аналізувати різні аспекти функціонування інформаційних та комунікаційних систем з використанням потужних інструментів [10].

Перелік інструментів, що входять до складу Java Modeling Tools (JMT):

- *JSIMgraph* - графічний інтерфейс для моделювання і аналізу дискретно-подійних систем. Дозволяє створювати, редагувати та візуалізувати моделі.
- *JMVA (Java Multithreaded Virtual Analyzer)* - інструмент для аналізу продуктивності мережевих систем, зокрема через аналіз масових обслуговувань.
- *JMT Queueing Network Solver* - розширення для JMVA, яке дозволяє моделювати мережеві системи через мережі черг і аналізувати їх продуктивність.
- *JMT Resolver* - інструмент для розв'язання систем з обмеженнями, які можуть виникати під час аналізу моделей.
- *JMT Plugins* - додаткові плагіни та розширення для роботи з JMT, які можуть включати в себе інші інструменти для аналізу та моделювання.

Переваги JMT полягають в його функціональності та можливостях, які він пропонує користувачам у галузі моделювання ІКС :

1. *Модульність* - JMT вражає своєю модульною структурою, що дає можливість розширювати його функціональність за потреби. Це означає, що користувачі можуть налаштовувати інструмент під конкретні вимоги свого дослідження або проекту.

2. *Різні методи моделювання* - підтримує різні методи моделювання, включаючи мережі Петрі, дискретно-подійне моделювання, інші математичні підходи та алгоритми. Це робить інструмент вельми гнучким у використанні та дозволяє його застосовувати для різноманітних досліджень.

3. *Графічний інтерфейс* - JMT надає користувачам зручний графічний інтерфейс, який спрощує процес створення, налаштування та редагування моделей ІКС. Це особливо важливо для користувачів, які не є програмістами або математиками.

4. *Аналіз продуктивності* - один із ключових аспектів JMT - це можливість проводити аналіз продуктивності ІКС. Користувачі оцінюють такі параметри, як пропускна здатність системи, середній час обслуговування, завантаженість ресурсів та інші. Це робить JMT важливим інструментом для проектування та оптимізації ІКС.

Java Modelling Tools є потужним та універсальним інструментом, який сприяє вивченню, аналізу та оптимізації інформаційно-комунікаційних систем. Він є важливим активом для дослідників та інженерів у галузі телекомунікацій, інформаційних технологій та інших сфер, де важливе моделювання та аналіз систем [1].

Порівнюючи JMT з іншими системами моделювання, можна визначити переваги та недоліки кожної системи.

1.3 Вибір і характеристика Java Modeling Tools (JMT) як інструменту для дослідження

На етапі проектування масштабного програмного забезпечення особливо важливим є питання моделювання продуктивності програмного забезпечення, оскільки саме це забезпечить можливість оптимізації продуктивності та управління складними програмами.

Java Modeling Tools (JMT) є інструментом, який візуально допоможе розробникам та архітекторам передбачити продуктивність системи та швидко відповісти на запитання "що-якщо". Це програмне забезпечення було розроблено з двома основними цілями:

- 1) підтримати вчених і практиків з оцінки продуктивності в аналізі складних систем;
- 2) як дидактичний інструмент, який допомагає студентам зрозуміти основні

принципи оцінювання та моделювання діяльності.

JMT має зручний додаток JSIMgraph, який вибирається через головний інтерфейс JMT (рисунок 1.4):

JSIMgraph – це графічне середовище проектування для мережевих моделей черги, яке тісно пов'язане з JSIMengine для моделювання дискретних подій. JSIMwiz замінює графічну структуру JSIMgraph набором майстрів, які направляють користувача через визначення моделі черги. Інструменти створюють XML-специфікації імітаційних моделей, друковану візуалізацію складних мереж, автоматичне налагодження моделі, підтримку аналізу «що-якщо» та динамічне представлення стану симуляції, оцінки показників продуктивності та відповідні довірчі інтервали. JSIMengine підтримує оцінку найпопулярніших типів моделей масового обслуговування та кількох конструкцій, які неможливо розв'язати за допомогою точних аналітичних методів, наприклад багатокласових мереж масового обслуговування з блокуванням, пріоритетами, елементами розгалуження та з'єднання, та схемами маршрутизації, що залежать від стану [11].

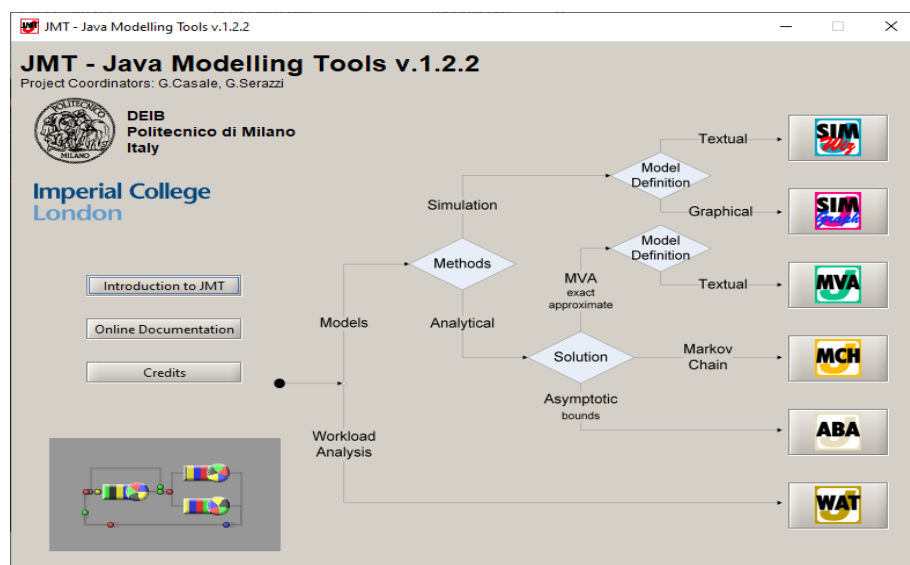


Рисунок 1.4 – Інтерфейс Java Modeling Tools

JSIMgraph має зручний інструмент для визначення структури мережі та завдання атрибутів, що дозволяє створювати моделі, які нелегко оцінити аналітично і, які потребують вимагають моделювання для аналізу [9].

Відтак цей засіб дають можливість моделювати розпаралелювання процесів, яке проводиться з метою балансування навантаження. Симулятор також підтримує області обмеженої ємності, що накладає обмеження на максимальну кількість завдань, які мають доступ до локальної підмережі серверів. Вони надзвичайно корисні для фіксації поведінки ефектів насичення продуктивності, які накладаються політиками контролю доступу або обмеженнями пам'яті [1].

Таким чином, дане програмне забезпечення дозволяє доволі ефективно проводити аналіз комп'ютерних мереж з різними складовими частинами та різними параметрами їх функціонування. Відкритий код дозволяє використовувати цей інструмент без обмежень.

Підсумуючи можна сказати, що першою і однією з ключових причин вибору JMT була його відкрита природа та безкоштовність. Цей інструмент є вільним програмним продуктом, що дозволяє без обмежень користуватися його можливостями. Відсутність ліцензійних витрат робить JMT доступним для використання в академічних дослідженнях та навчальних цілях.

Другою важливою причиною є функціональність JMT - інструмент надає широкий набір можливостей для моделювання та аналізу ІКС. Зокрема, JMT підтримує дискретно-подійне моделювання, системи масового обслуговування, мережі Петрі, що дозволяє досліджувати різні аспекти функціонування систем. Ще однією важливою перевагою JMT є його графічний інтерфейс, зокрема тула Jsimgraph. Цей інтерфейс спрощує створення та редагування моделей, що особливо важливо для користувачів без глибоких технічних навичок у програмуванні чи математичному моделюванні [5].

Аргументи на користь вибору Java Modelling Tools (JMT) як інструменту для дослідження:

- *Універсальність* - JMT є універсальним інструментом, придатним для моделювання різних видів інформаційно-комунікаційних систем (ІКС). Він може бути успішно використаний для аналізу телекомунікаційних систем, комп'ютерних мереж, веб-серверів та інших об'єктів. Така універсальність

дозволяє досліджувати та моделювати різноманітні аспекти ІКС без необхідності використання різних інструментів.

- *Розширюваність* - дозволяє розширювати його функціональність залежно від конкретних потреб користувача. Це означає, що користувачі можуть створювати власні модулі та розширення для адаптації ЈМТ до конкретних завдань дослідження. Такий підхід дозволяє налаштовувати інструмент для вирішення конкретних завдань та враховувати специфіку об'єктів дослідження.

- *Активна спільнота* - ЈМТ має активну та розвинуту спільноту користувачів і розробників. Це забезпечує підтримку інструменту та його постійний розвиток. Користувачі можуть отримувати допомогу, консультації та рекомендації від інших членів спільноти, що сприяє ефективному використанню ЈМТ та розв'язанню можливих проблем.

В цілому, ці аргументи підкреслюють доцільність вибору ЈМТ як інструменту для дослідження та моделювання ІКС, забезпечуючи його гнучкість, універсальність та підтримку завдяки активній спільноті [5].

2 ДОСЛІДЖЕННЯ ОСНОВНИХ МЕХАНІЗМІВ ФУНКЦІОНУВАННЯ СКЛАДНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

2.1 Моделювання системи в середовищі ЈМТ на прикладі розробки моделі веб-серверу

2.1.1. Опис процесу створення імітаційної моделі

Будь-яку обчислювальну систему можна розглядати як набір ресурсів (апаратного та програмного забезпечення), які виконують запити на обробку, подані користувачами. Тому вхідні параметри моделі можна розділити на дві групи відповідно за навантаженням і ресурсами. Залежно від системи, що моделюється, у подальшому запити на обробку будуть взаємозаміно називатися завданнями, програмами, клієнтами, запитами або користувачами, а ресурси будуть називатися станціями, елементами, компонентами або центрами обслуговування [1].

Запити, що надходять, разом називають робочим навантаженням, характеристика робочого навантаження відноситься до їх кількісного опису. Якщо окремі компоненти робочого навантаження мають подібні характеристики, вони групуються разом, а їхні статистичні параметри, такі як середнє значення, стандартне відхилення та розподіл, використовуються як вхідні дані для моделей. У цьому випадку навантаження називають однорідним або однокласним, а моделі називають моделями одного класу. Компоненти робочого навантаження, що складається з різних типів програм, як правило, мають суттєво різні вимоги до обслуговування. У цьому випадку необхідно ідентифікувати кілька груп компонентів зі схожими характеристиками, а робоче навантаження називають неоднорідним або множинним класом (мультикласом). Кожен клас описується зі своїми статистичними характеристиками [1].

Подальше дослідження будемо проводити для інформаційно-комунікаційної системи. Модель системи складається з Інтернет сервера (рисунок 2.1).

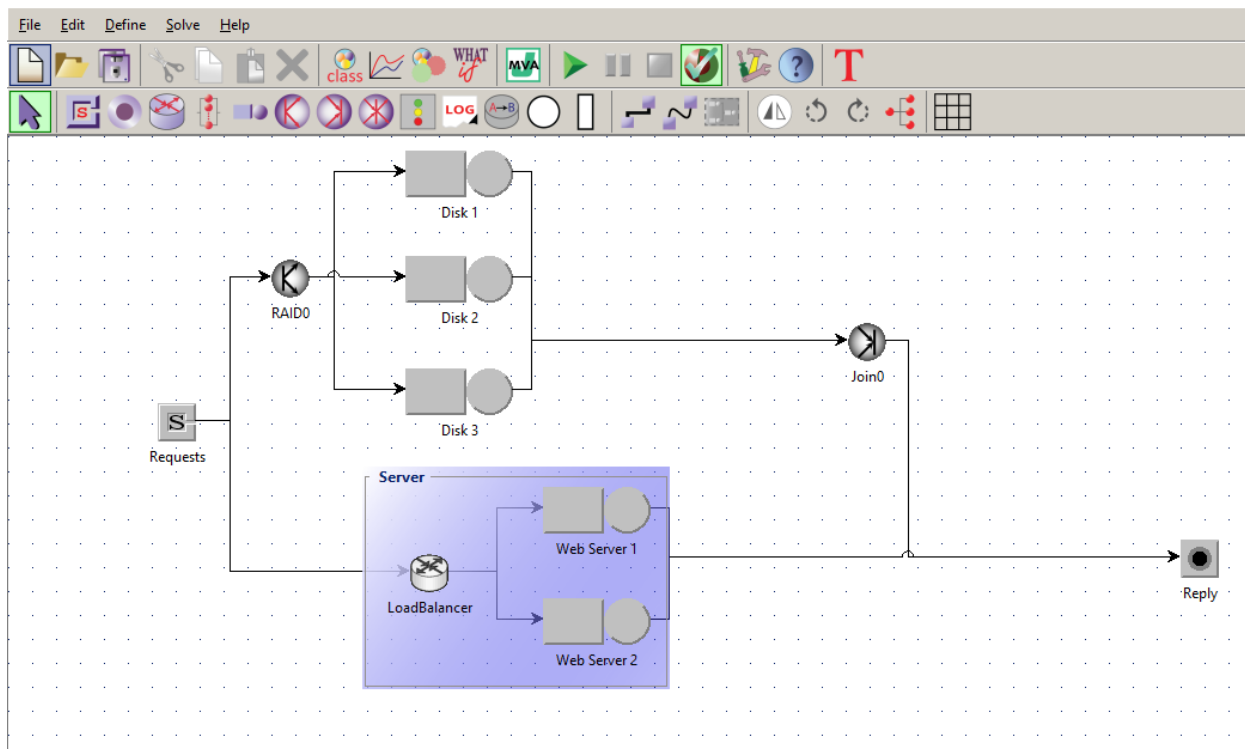


Рисунок 2.1 – Структура моделі Інтернет сервера

Дана система складається з двох серверів, для підвищення їх продуктивності

вводиться обмеження на одночасну обробку декілька сотень запитів. Для забезпечення надійності роботи сервера передбачається використовувати RAID0 масив, який складається з 3 однотипних дисків.

Для забезпечення підвищеної відмовостійкості сервера передбачається використовувати балансування навантаження. Це дозволяє розподілити завдання між кількома мережевими пристроями (в нашому випадку 2 серверами) для оптимізації використання ресурсів, скорочення часу обслуговування запитів [12].

Перейдемо до встановлення параметрів системи. Будемо вважати, що система передбачає два види запитів. Запити першого типу мають доступ лише до сервера, запити другого типу відвідують сховище або сервер відповідно до принципу обов'язкового обслуговування. Запити, які поступають до масиву RAID0 випадковим чином між 3 дисками.

Опишемо елементи ЖТ, які використовуються для побудови моделі.

Запити, що надходять, разом називають робочим навантаженням, характеристика робочого навантаження відноситься до їх кількісного опису. Робочі навантаження можуть бути двох типів: відкриті та закриті. Коли робоче навантаження відкрите, кількість клієнтів у моделі коливається та може нескінченно зростати, якщо станція стає насиченою, тоді як із закритим навантаженням це число залишається постійним. Прикладом відкритого робочого навантаження є потік запитів, що надходять із з'єднання з Інтернетом, який зазвичай продукує доволі багато запитів. Прикладом закритого робочого навантаження є обчислювальна інфраструктура, до якої можуть отримати доступ лише працівники компанії. Кількість клієнтів фіксована і обмежується співробітниками компанії. Відтак, коли досягнута максимальна кількість клієнтів, яка може одночасно виконуватися, новий клієнт може увійти в систему лише тоді, коли клієнт завершить своє виконання. Моделі, які використовують відкриті робочі навантаження, називаються відкритими моделями, тоді як ті, що виконують закриті робочі навантаження, називаються закритими моделями [1].

У імітаційних моделях з відкритими робочими навантаженнями клієнти, які надходять до системи, генеруються вихідною станцією, а в кінці виконання направляються до приймальної станції (у JSIMgraph також станції Fork, Class Switch і Transition станції можуть генерувати клієнтів). Також можливі моделі з обома типами відкритих і закритих робочих навантажень, що працюють одночасно, і називаються змішаними моделями [1].

Станція, яка використовується для обчислення індексів продуктивності на рівні системи (пропускна здатність, час відгуку, глобальне використання ресурсів тощо), називається опорною станцією (RS). Коли завдання проходить через RS, у більшості випадків неявно передбачається, що його виконання завершено, і тому воно залишає систем. У цьому випадку завдання відвідує RS лише один раз протягом свого життя, і з цієї причини станція затримки, яка в більшості моделей імітує користувачів, часто називається як RS. Проте будь-який із компонентів моделі може бути обраний як RS. Зрозуміло, що цей вибір впливає на всі показники ефективності. Щоб обчислити їх значення, відвідування кожного компонента моделі необхідно масштабувати відносно відвідувань RS. У відкритих моделях JSIMgraph за замовчуванням передбачає вихідну станцію як RS. Коли завдання завершує своє виконання, індекси його продуктивності обчислюються з урахуванням інтервалу часу, що минув між його створенням із джерела та виходом із приймача. В закритих моделях будь-яку станцію можна вибрати як RS. У цьому випадку її індекси продуктивності розраховуються з урахуванням часу, що минув між його генерацією з RS і входом в модель, і часом, коли він виходить з моделі і досягає RS [1].

У моделі компоненти, що представляють системні ресурси, можуть бути різних типів. Компоненти, які найчастіше використовуються в аналітично розв'язаних моделях, зазвичай мають два типи: черга та затримка. У компонентах черги запити надходять, змагаються за сервер, чекають у черзі, коли він зайнятий, виконуються, коли він стає неактивним, а потім виходять. Компонент затримки просто вводить затримку в потік запитів, але черга не створюється. У цьому випадку запит, що надійшов, завжди знаходитиме

неактивний сервер, оскільки вважається, що вони нескінченні. Багато типів компонентів використовуються в симуляційних моделях залежно від розглянутого інструменту та складності системи, що моделюється. Наприклад, у JSIMgraph станції Fork і Join використовуються для моделювання паралелізму, FCR Finite Capacity Regions для контролю доступу до модельних регіонів, Semaphore для вибіркового блокування запитів[14].

Отже, для моделювання нашої системи пропонується встановити такі загальні характеристики: мультикласова відкрита система.

Для моделювання сервера використовується FCR (Finite Capacity Region). Для моделювання алгоритму RAID0 використовується станція Fork. Коли запит надходить на станцію Fork RAID0, він розбивається на три запити, які надсилаються паралельно на всі диски сервера [13].

Для моделювання дисків та серверів використовується елемент черги (Queue).

2.1.2. Таблиці з характеристиками елементів системи.

Перейдемо до опису параметрів моделі.

Визначимо класи системи (рисунок 2.2). Для кожного класу потрібно визначити його тип та закон розподілу, за яким він поступає в систему (таблиця 2.1).

Таблиця 2.1 – Параметри класів

Колір	Назва	Тип	Пріоритет	Закон розподілу
Синій	Class0	відкритий	0	гіперекспоненціальний (p, λ_1, λ_2)=(0.291,0.182,0.444)
Червоний	Class1	відкритий	0	експоненціальний $\lambda=5$

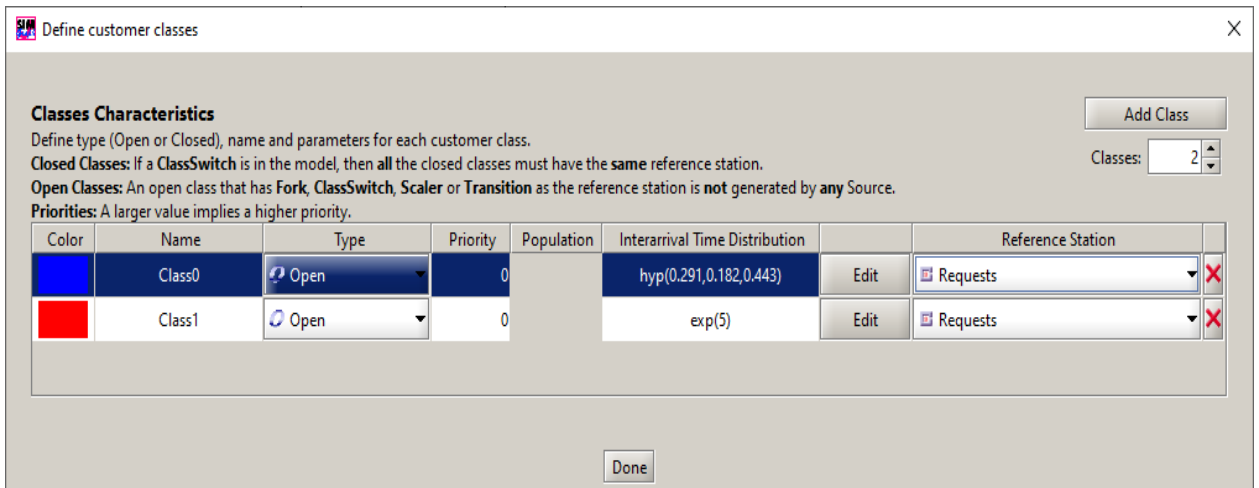


Рисунок 2.2 – Визначення класів системи

Усі клієнти Class0 надходять в систему з гіперекспоненціальним розподілом з параметрами $(p; 1; 2) = (0,291, 0,182, 0,442)$, в результаті чого середнє значення дорівнює 3,2 с. і коефіцієнт варіації 1,192 (рисунок 2.3).

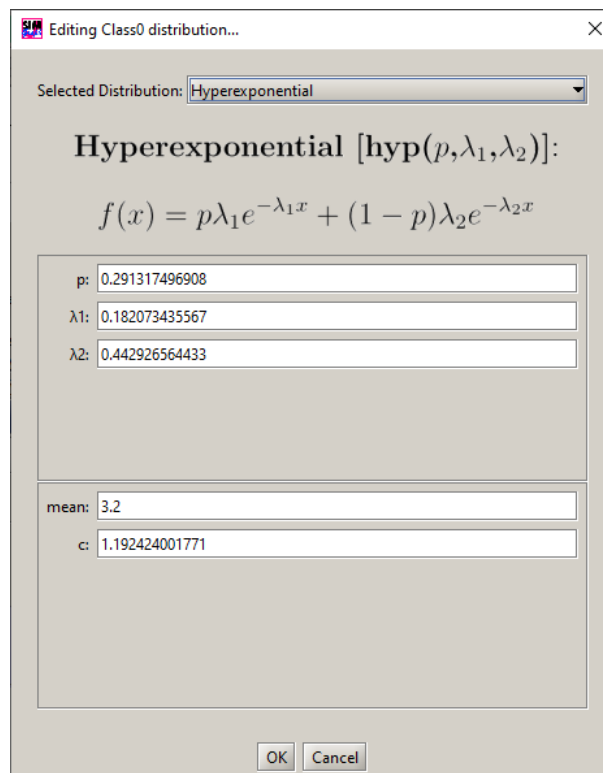


Рисунок 2.3 – Параметри розподілу для Class0

Усі клієнти Class1 надходять в систему з експоненціальним розподілом з параметром $\lambda=5$, в результаті чого середнє значення дорівнює 0,2 с (рисунок 2.4).

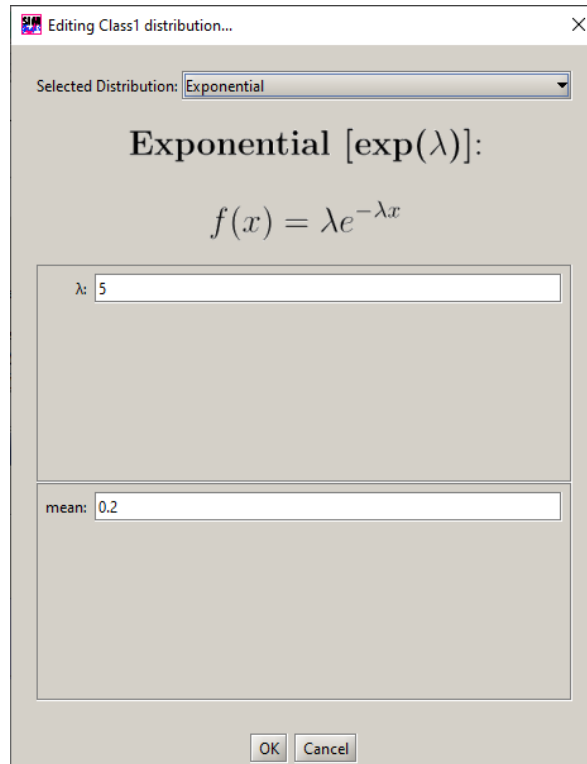


Рисунок 2.4 – Параметри розподілу для Class1

Розглянемо більш детально формули які були визначенні як закон розподілу для створених класів:

1. Для Class0 була використана формула гіперекспоненціального розподілу (див. рис. 2.3).

Гіперекспоненціальний розподіл - це комбінація декількох експоненціальних розподілів з різними параметрами. Цей розподіл описується за допомогою функції розподілу й ймовірностей.

У цій формулі:

- p - ймовірність вибору компоненти,

- λ_1, λ_2 - параметри експоненціальних розподілів, які визначають обернені середні для компоненти.
2. Для Class1 була використанна формула експоненціального розподілу (див. рис. 2.4)

Експоненціальний розподіл - це статистичний розподіл, який описує час між подіями в процесі Пуассона, тобто час очікування до настання події.

У цій формулі:

- x - змінна, що представляє собою час між подіями,
- λ - параметр інтенсивності (частоти) подій.

Експоненціальний розподіл часто використовується для моделювання часу між подіями, таких як інтервали між збоями в системі та час між приходами клієнтів.

Взагалі, симулятор підтримує різні розподіли для процесів прибуття та обслуговування, включаючи експоненціальний, Ерланг, та гіперекспоненціальний, Парето та гамма. Вони отримуються, як зазвичай, шляхом оцінки оберненої кумулятивної функції розподілу (CDF) для випадкового числа, відібраного з рівномірного розподілу. Високоякісні псевдовипадкові числа генеруються за допомогою рушія Mersenne Twister, який має низькі обчислювальні вимоги [11].

У JSIMgraph користувачеві надається гнучкість у визначенні залежних від навантаження тарифів обслуговування, пов'язуючи інший розподіл послуг із певним діапазоном населення в локальній черзі. Крім того, параметри кожного розподілу можна виразити як довільну математичну функцію поточної сукупності [11].

Перейдемо до визначення параметрів інших елементів моделі.

Елемент Requests приймає два класи запитів, для яких визначаються різні стратегії маршрутизації (рисунок 2.5). Оскільки потік запитів Class0 не повинен потрапляти в масив RAID, то для нього встановлюється ймовірність для

маршруту балансувальник навантаження (Load Balancer) 1, а для маршруту RAID ймовірність дорівнює 0 (таблиця 2.2). Для потоку запитів Class1 встановлена стратегія Round Robin (клієнти циклічно направляються до вихідних каналів, перший клієнт надсилається на RAID, другий клієнт надсилається на RAID. Наступний клієнт знову відправляється на RAID і так далі).

Таблиця 2.2 – Параметри елемента Requests

Назва класу	Стратегія маршрутизації	Параметри
Class0	Probabilities	Load Balancer = 1 RAID = 0
Class1	Round Robin	-

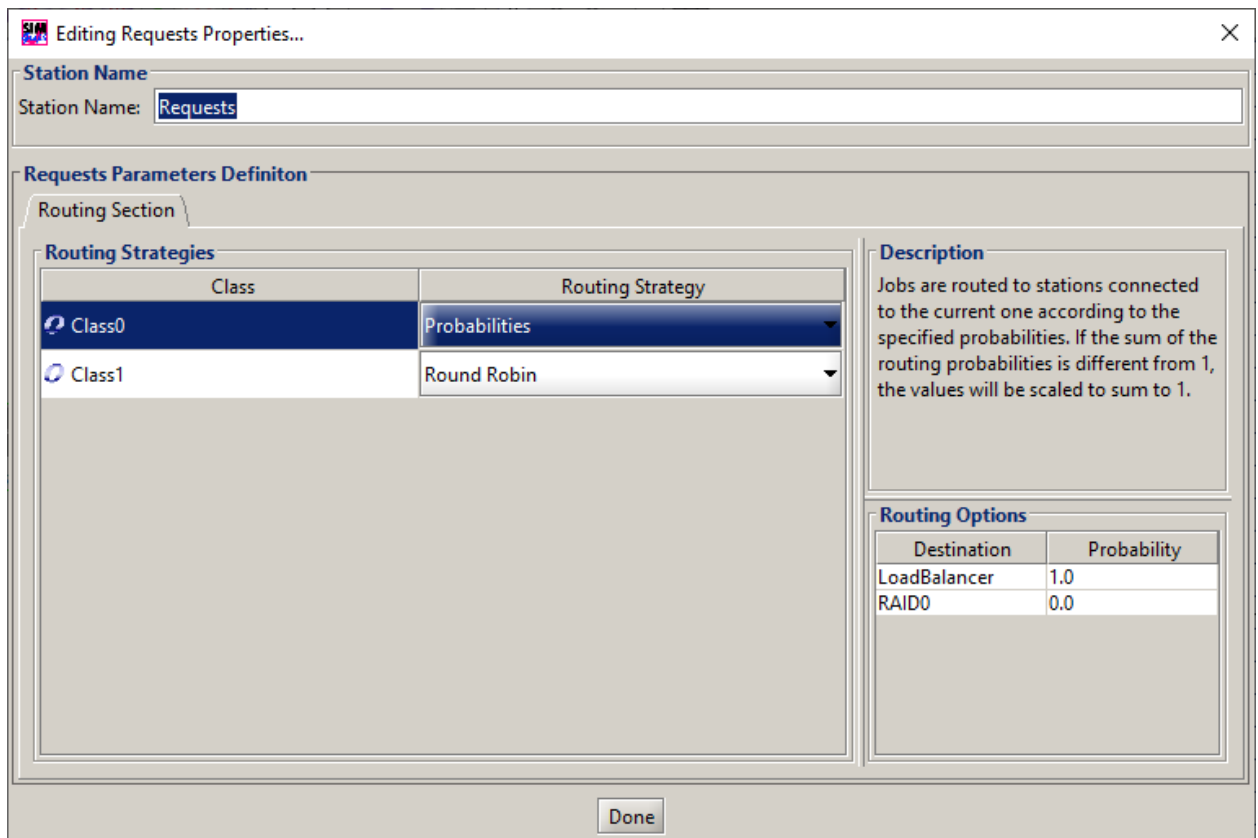


Рисунок 2.5 – Встановлення стратегії маршрутизації

Для моделювання RAID0 використовується станція Fork, яка розділяє завдання на кілька завдань, які виконуються паралельно. Час обслуговування не виділено для цієї операції, тому немає запиту специфікації часу обслуговування. Завдання направляються за вихідними посиланнями Fork. Параметри станції Fork визначаються 3 секціями: Capacity Section, Queue Section, Fork Strategies.

У секції Queue Section(рисунок 2.6) можна вказати ємність черги (кінцева або безкінечна) і політику планування.

Ємність – станція може прийняти будь-яку кількість клієнтів і залишити їх у черзі, у цьому випадку її ємність вважається безкінечною [12].

Політика черги визначає алгоритм, який використовується для прийняття рішення, якого клієнта обслуговувати наступним. Різноманітні фактори можуть впливати на порядок обслуговування клієнтів, наприклад порядок прибуття, пріоритети, пов'язані з класом, кількість послуг, які вже надаються клієнтам, тощо [12].

Дисципліни планування – Невипереджувальне планування. Також встановлена політика черги – FCFS: клієнти обслуговуються в тій самій послідовності, в якій вони прибувають на станцію.

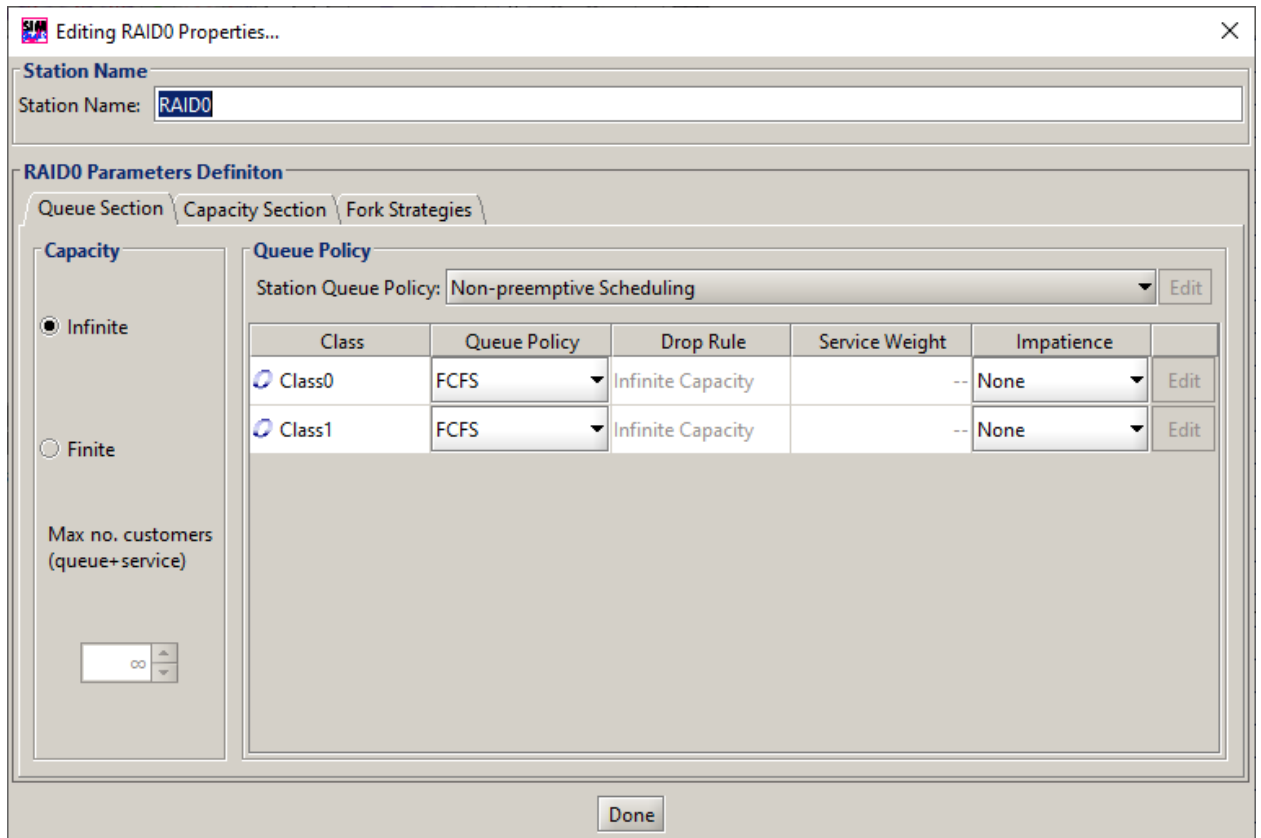


Рисунок 2.6 – Встановлення параметрів для станції RAID0

В секції Fork Strategies встановлюємо стратегію за замовчуванням. У цьому випадку кількість завдань, створених у кожній вихідній ланці (називається ступенем розгалуження), для кожної роботи, що надходить на станцію Fork, однакова. В нашому випадку кількість завдань дорівнює 3.

Для моделювання дисків використовуємо станцію Queue. Основні параметри цих станцій наведені в таблиці 2.3.

Таблиця 2.3 – Основні параметри станцій Диск

Назва станції	Потік запитів	Закон розподілу обслуговування	Політика черги	Стратегія маршрутизації
Диск 1 (Disk1)	Class0	Load Independent Експоненціальний $\lambda=5$	FCFS	Random
	Class1	Load Independent Експоненціальний $\lambda=0.5$	FCFS	Random

Диск (Disk1)	2	Class0	Load Independent Експоненціальний $\lambda=5$	FCFS	Random
		Class1	Load Independent Експоненціальний $\lambda=0.5$	FCFS	Random
Диск (Disk1)	3	Class0	Load Independent Експоненціальний $\lambda=5$	FCFS	Random
		Class1	Load Independent Експоненціальний $\lambda=0.5$	FCFS	Random

У розділі маршрутизації для кожного класу клієнтів обраний параметр Random, що визначає шлях, за яким завершені завдання направляються до пристроїв, підключених до станції, для якої визначено стратегію маршрутизації.

Встановлена політика черги – FCFS, тобто обслуговується перший клієнт, який прийшов першим.

В секції Service Section (рисунок 2.7) встановлюється стратегія обслуговування "незалежно від навантаження". Незалежно від кількості клієнтів, які знаходяться на станції, система обслуговуватиме всіх клієнтів, дотримуючись фіксованої політики, змодельованої вибраним статистичним

розподілом. Статистичний розподіл для потоку Class0 – експоненціальний з параметром $\lambda=5$, для потоку Class1 – експоненціальний з параметром $\lambda=0.5$

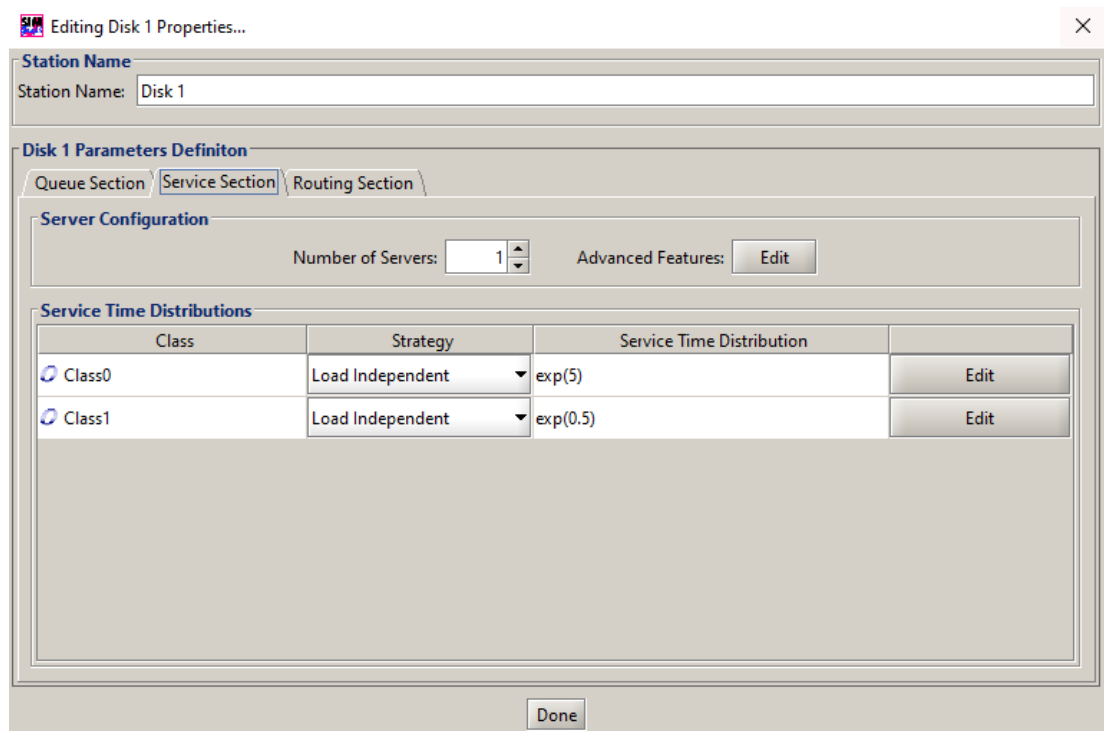


Рисунок 2.7 – Приклад встановлення параметрів обслуговування черги для станції Disk1

Для моделювання сервера використовується FCR(рисунок 2.8).

Його основним параметром є ємність, яку за попередніми умовами визначили в 100 од.

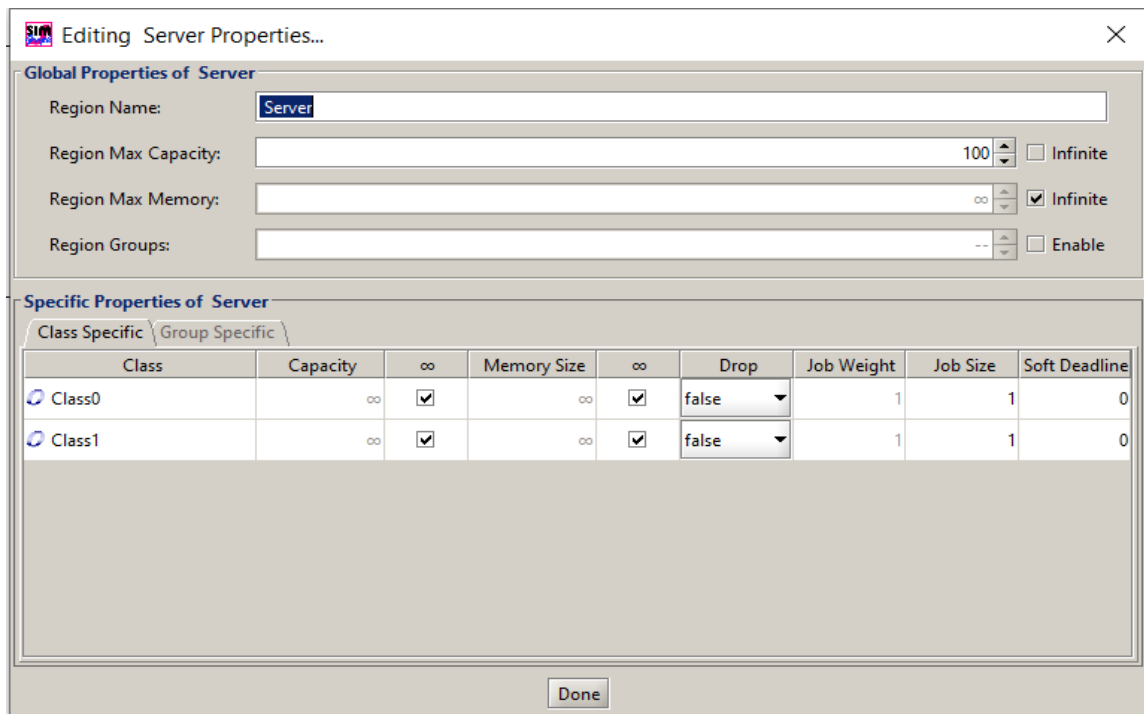


Рисунок 2.8 – Параметри FCR в моделі

Встановлені параметри визначено такі:

Ємність – максимальна кількість клієнтів цього класу в FCR (безкінечно, якщо обмеження не встановлено).

Розмір пам'яті – максимальний розмір пам'яті, зайнятої завданнями цього класу в регіоні (безкінечно, якщо обмеження не встановлено)

Drop – коли FCR зайнятий і Drop має значення false, вхідні завдання не відкидаються, а ставляться в чергу за межами регіону, доки FCR не зможе їх прийняти. Якщо Drop має значення true, завдання видаляються із системи.

Для механізму балансування навантаження були встановлені параметри, зображені на рисунку 2.9.

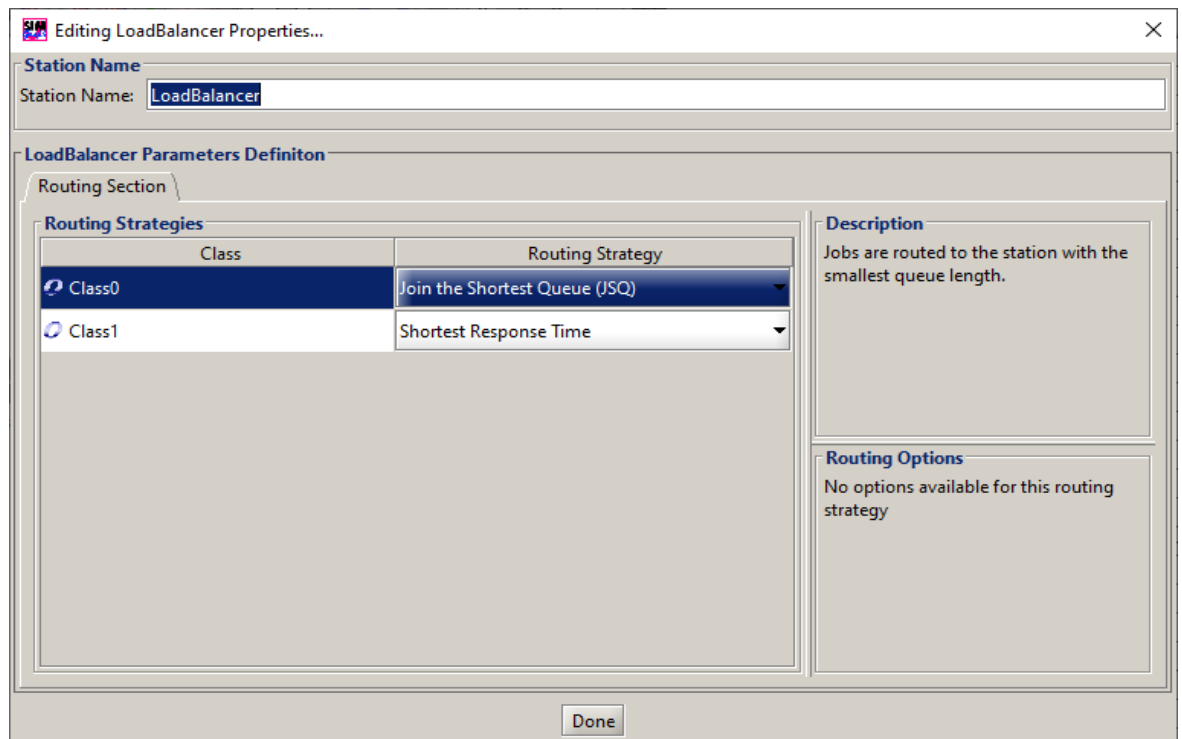


Рисунок 2.9 – Параметри механізму балансування навантаження

Для Class0 встановлено Join the Shortest Queue (JSQ) (найкоротша довжина черги). Кожен клієнт направляється до підключеної на виході станції, яка має найменшу кількість клієнтів у черзі та в обслуговуванні на момент, коли клієнт залишає станцію маршрутизації [14].

Для Class1 встановлено Shortest Response time (найкоротший час відповіді). Клієнти надсилаються на станцію, де час відповіді для класу завдання є найменшим у момент, коли клієнт залишає станцію маршрутизації [14].

Нарешті, для двох Web-серверів встановлені параметри, наведені в таблиці 2.4.

Таблиця 2.4 – Основні параметри станцій сервер

Назва станції	Потік запитів	Закон розподілу обслуговування	Політика черги	Стратегія маршрутизації
Web-server1	Class0	Load Independent Експоненціальний $\lambda=1$	FCFS	Random
	Class1	Load Independent Експоненціальний $\lambda=1$	FCFS	Random
Web-server2	Class0	Load Independent Експоненціальний $\lambda=1$	FCFS	Random
	Class1	Load Independent Експоненціальний $\lambda=1$	FCFS	Random

3 АНАЛІЗ ФУНКЦІОНУВАННЯ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНОЇ СИСТЕМИ НА ПРИКЛАДІ ВЕБ-СЕРВЕРУ

3.1 Опис результатів моделювання

Після встановлених параметрів моделі перейдемо до визначення параметрів симуляції(рисунок 3.1).

Симуляційне початкове число – це число, яке використовується системою моделювання для генерації псевдовипадкових чисел. При виборі випадкового значення симулятор використовуватиме початкове число, згенероване випадковим чином. Коли моделювання повторюється з використанням того самого початкового числа, буде згенерована та сама послідовність псевдовипадкових чисел, що призведе до ідентичних результатів [1].

Максимальна тривалість (час роботи, секунди) – це максимальний час у секундах, протягом якого виконуватиметься моделювання, тобто значення часу при моделюванні. Якщо симуляція закінчується раніше максимального часу (через автоматичне керування зупинкою), цей параметр ігнорується. Значення за замовчуванням – нескінченність. Можна встановити власне значення, тоді симуляція зупиниться, навіть, якщо не буде встановлено стійке рішення [11].

Максимальний змодельований час – це максимальний змодельований час, дозволений для поточного запуску симуляції. Симуляція буде зупинена, коли цей час буде досягнуто (якщо інші критерії зупинки, які активовані, не досягнуті) [11].

Максимальна кількість зразків – це максимальна кількість зразків (даних) для кожного індексу, який збирає JSIMgraph перед завершенням моделювання. Значення за умовчанням для максимальної кількості зразків становить 1 000 000; його можна збільшити (для точнішого моделювання) або зменшити (для швидшого моделювання) [11].

Інтервал оновлення анімації (сек) – це деталізація, з якою результати відображаються на екрані, тобто інтервал часу між графіком двох послідовних значень графіків у процесі моделювання. Прапорець анімації використовується, щоб увімкнути або вимкнути анімацію сюжету під час виконання симуляції [11].

Оскільки модель є відкритою, то при моделюванні можна задати кількість клієнтів, що знаходиться в тій чи іншій станції. Для проведення моделювання залишимо нульове значення без змін(див. рис. 3.1).

Simulation Parameters
Define simulation parameters and initial customer locations.

Simulation random seed: random

Maximum duration (sec): infinite

Maximum simulated time: infinite

Maximum number of samples: no automatic stop

Maximum number of events: infinite

Animation update interval (sec): animation

Initial customer locations:
Jobs located in **Router, Fork, Logger** and **ClassSwitch** are treated as arriving into these stations.

*	Web Server 1	Web Server 2	LoadBalancer	Disk 1	Disk 2	Disk 3	RAID0
Class0	0	0	0	0	0	0	0
Class1	0	0	0	0	0	0	0

Done

Рисунок 3.1 – Параметри імітації.

Основні показники змодельованої системи:

- Швидкість відключення (станції або всієї системи) – це власне швидкість, з якою клієнти відключаються від станції або регіону через виникнення обмеження (наприклад, максимальна місткість черги, максимальна

кількість клієнтів у регіоні). На рівні системи це стосується швидкості, з якою клієнти виключаються із системи.

- FCR Memory Occupation описує зайнятість основної пам'яті області обмеженої ємності. Загальна вага FCR описує загальну вагу клієнтів, обслужених у регіоні кінцевої ємності [11].
- Пропускна здатність (перехідної станції) – швидкість, з якою перехідна станція працює, тобто кількість дзвінків, що виникають за одиницю часу.
- Кількість клієнтів Fork Join – середня кількість клієнтів (завдань) у регіоні Fork/Join, які як чекають поза межами регіону, так і залишаються всередині регіону[14].
- Час відповіді на приєднання до розгалуження – середній час, проведений клієнтом (робочим завданням) у регіоні розгалуження/приєднання. Включено час синхронізації завдань на Join [11].
- Кількість клієнтів (станції чи всієї системи) – її можна визначити на рівні станції чи системи. На рівні станції це стосується як клієнтів, які чекають у черзі, так і тих, хто отримує обслуговування. На рівні системи цей показник показує глобальну кількість клієнтів у системі. Ці значення описані для кожного класу клієнтів [11].
- Час у черзі (на станції) – середній час, витрачений клієнтами на очікування в черзі на станції. Він не включає час обслуговування [15].
- Час перебування (на станції) – загальний час, проведений клієнтом на станції, як у черзі, так і при отриманні послуги, враховуючи всі відвідування станції, здійснені протягом повного виконання [11].
- Час відгуку (станції або всієї системи) – його можна визначити на рівні станції або системи. На рівні станції це означає середній час, проведений клієнтом на цій станції за одне відвідування (сума часу в черзі та часу обслуговування). На рівні системи це означає середній час, який клієнт проводить у системі, щоб отримати послугу від різних станцій, які він відвідує. Це відповідає інтуїтивному поняттю часу відповіді, як інтервалу між поданням запиту та отриманням відповіді. Його можна отримати,

підсумувавши час перебування всіх ресурсів. Ці значення описані для кожного класу клієнтів.

- Потужність системи (на рівні системи та на рівні класу) – оптимальна робоча точка системи – це точка, що відповідає максимальній пропускну здатності системи X з мінімальним часом відповіді системи R , тобто значення співвідношення X/R є максимальним в цьому пункті. Даний коефіцієнт відомий як системна потужність. Зазвичай обчислюється на рівні всієї системи. Потужність системи також може бути обчислена для кожного класу клієнтів, враховуючи пропускну здатність і час відгуку класу (у цьому випадку це називається системною потужністю на клас) [11].
- Пропускна здатність (станції або всієї системи): на рівні станції це означає швидкість, з якою клієнти відправляються зі станції, тобто кількість запитів, виконаних за одиницю часу. На рівні системи це стосується швидкості, з якою клієнти залишають систему. Ці значення описані для кожного класу клієнтів.
- Пропускна здатність на приймача: швидкість, з якою клієнти залишають систему щодо вибраного приймача, тобто кількість запитів, виконаних за одиницю часу, які досягають певного приймача. Ці значення описані для кожного класу клієнтів [11].
- Використання (станції): відсоток часу, протягом якого станція використовується (тобто зайнята), оцінюється протягом усього циклу моделювання. Він коливається від 0 (0%), коли станція завжди неактивна, до максимуму 1 (100%), коли станція постійно зайнята обслуговуванням клієнтів протягом усього циклу моделювання. Станції обслуговування можуть мати більше одного сервера, їх кількість є параметром, який потрібно вказати (за замовчуванням 1). Важливо зазначити, що коефіцієнт використання U станції масового обслуговування з одним сервером визначається як $U = S$, а на станції з m -серверами коефіцієнт використання будь-якого окремого сервера визначається як $U = S/m$. На станціях із

затримкою, для узгодженості із законом Літтла, використання обчислюється як середня кількість клієнтів на станції, і тому воно може бути більше 1.

Проведене моделювання показує постійну роботу елементів системи (рисунок 3.2).

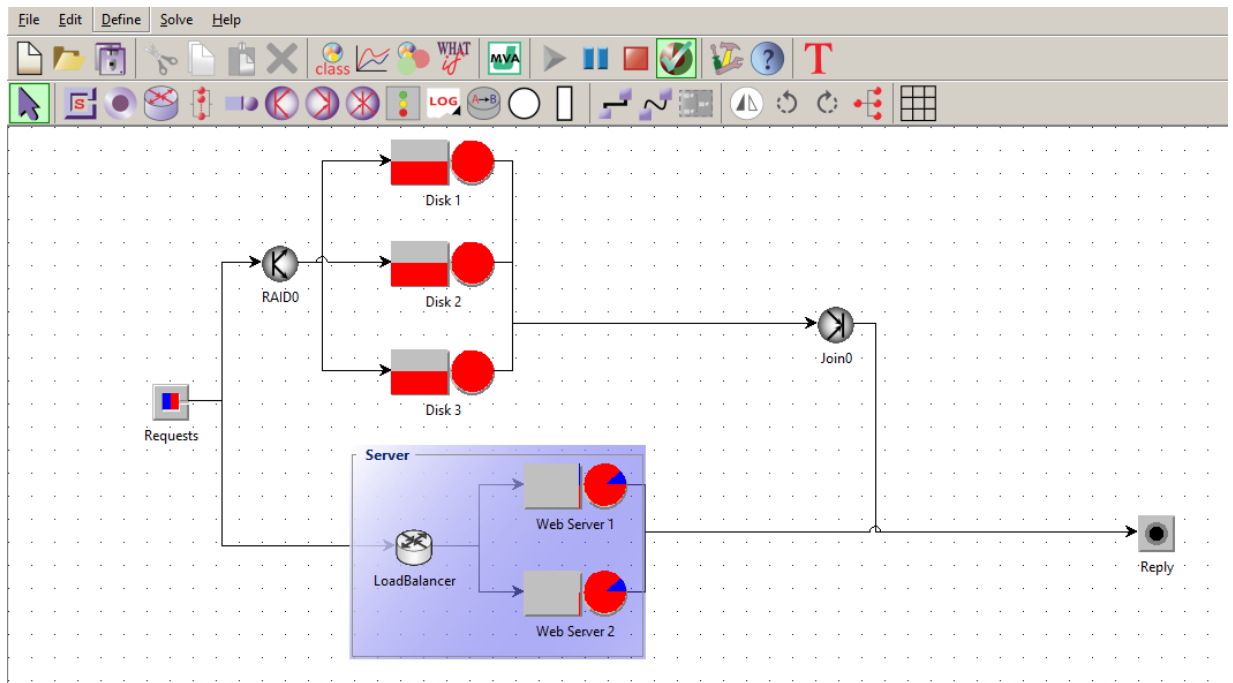
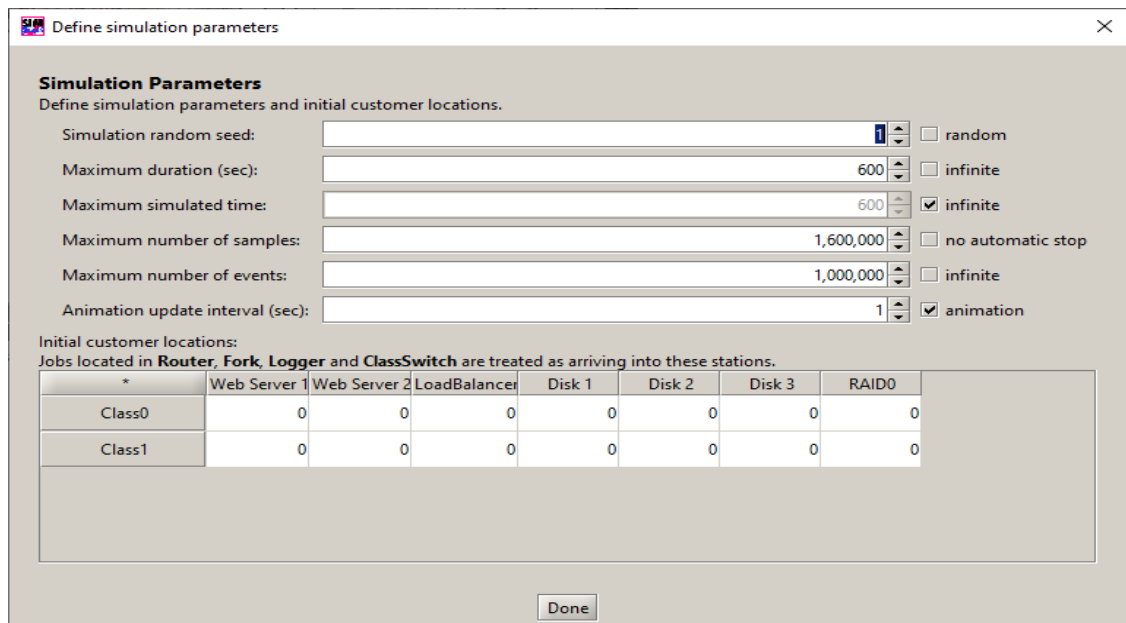


Рисунок 3.2 – Стан системи при моделюванні

Подальше моделювання з використанням в якості зупинки точки досягнення рівня значущості показало, що система є не стійкою, а її повний аналіз потребує

значних обчислювальних ресурсів. Тому було вирішено змінити параметри моделювання (рисунок 3.3).

Рисунок 3.3 – Зміна параметрів моделювання



Результати вимірювання різних параметрів системи наведені на рисунках 3.4-3.9.

У системі буде проводитись моделювання таких параметрів як час відповіді системи(див. рис. 3.4), пропускна здатність системи(див. рис. 3.5) та деяких її елементів(див. рис. 3.6), часу відгуку Web-server 1 (див. рис. 3.7), часу очікування в черзі системи (див. рис. 3.8), та її потужність(див. рис. 3.9).

В результатах моделювання(див. рис. 3.4 – 3.9), доступні наступні параметри:

- *Station name*: Це назва станції, для якої обчислюється індекс;
- *Class Name*: Це назва класу, для якого обчислюється індекс на станції ;
- *Conf.In / Max Rel.Err.*: Довірчий інтервал та максимальна відносна похибка обчисленого індексу;
- *Analyzed Samples*: Це кількість вибірок (зібраних даних), використаних для обчислення продуктивності;
- *Min.*: Це мінімальне значення, яке спостерігається для індексу;
- *Max*: це максимальне значення, яке спостерігалось для індексу;

Average value: Це розраховане середнє значення індексу, зазвичай це значення, що становить найбільший інтерес.

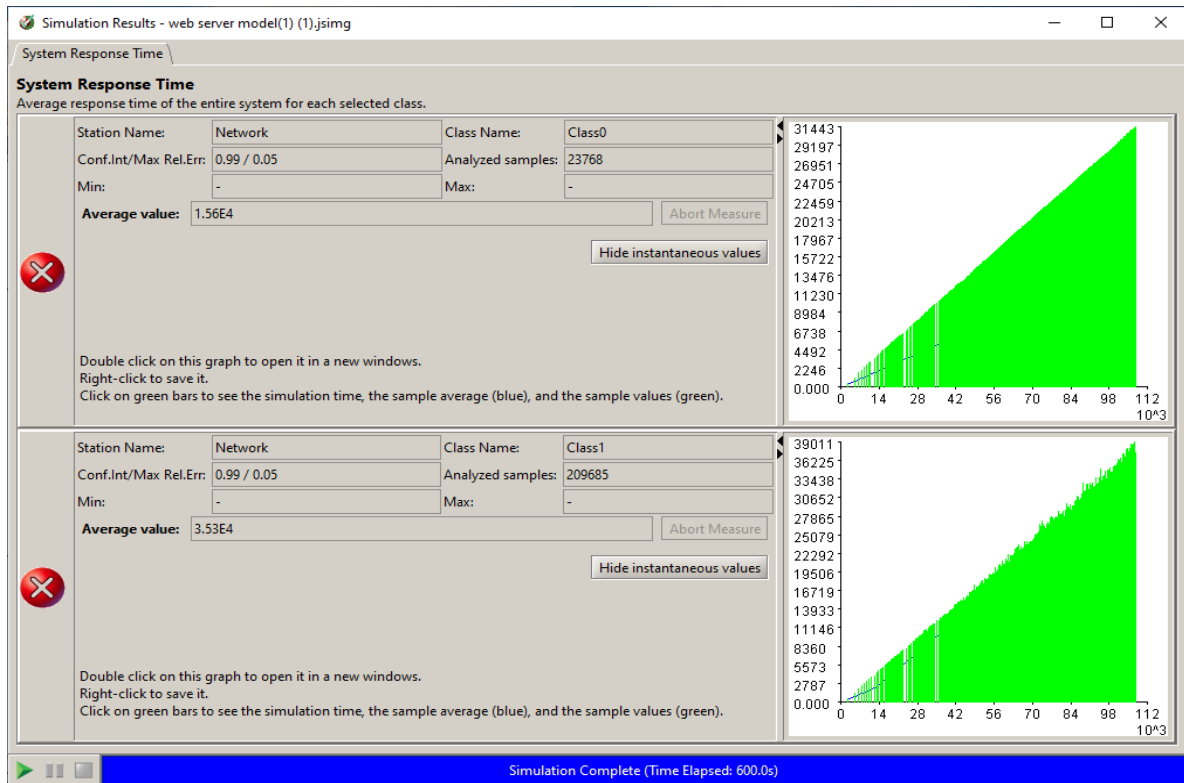


Рисунок 3.4 – Моделювання часу відповіді системи

З графіків моделювання (див. рис. 3.4) видно, що середній час відповіді системи (синя лінія) нескінченно зростає. Це пов'язано з тим що в систему надходить дуже велика кількість запитів і система фізично не встигає обробити їх всіх. Як наслідок запити скупчуються, і очікують часу звільнення системи або існуючи ігноруються і система починає обробляти інші. Також, це пов'язано з обмеженою пропускною здатністю системи.

Спираючись на це, розглянемо результати моделювання пропускної здатності системи (див. рис. 3.5).

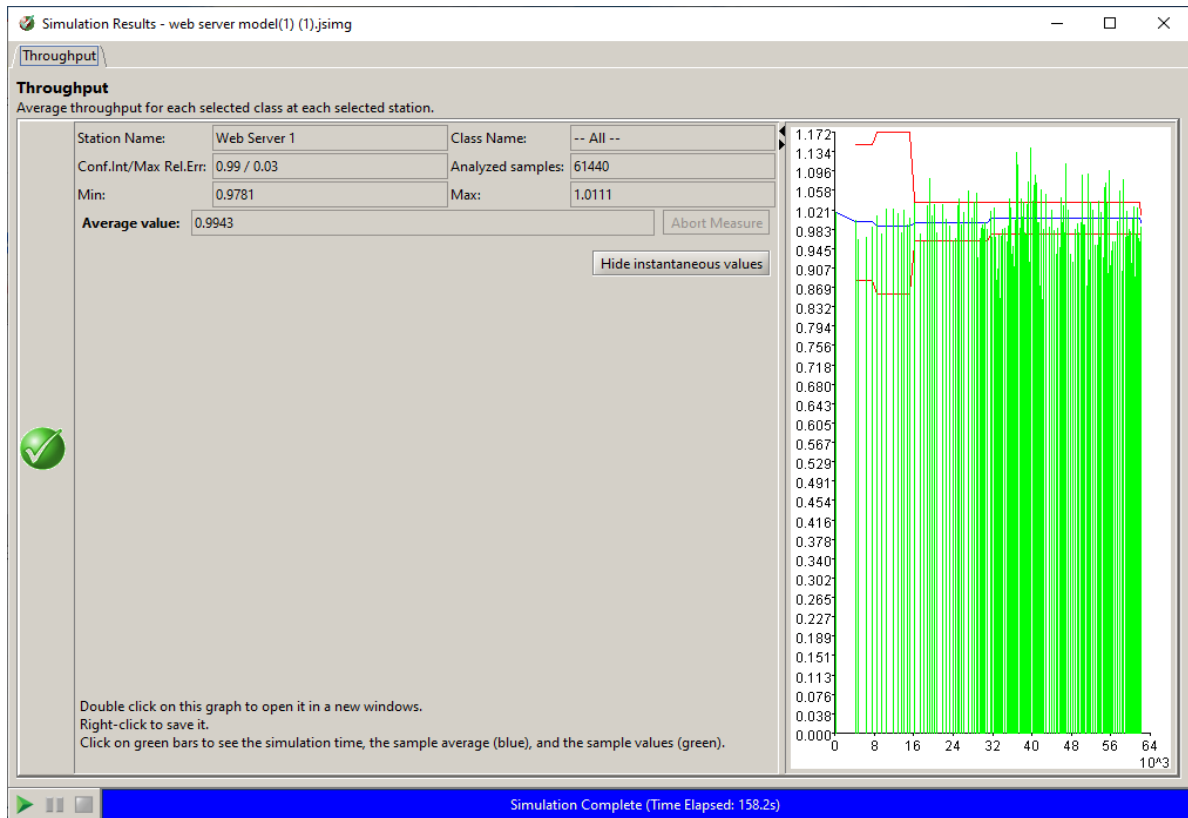


Рисунок 3.5 – Моделювання пропускної здатності системи

На графіку видно пропускну здатність, з якою клієнти виходять зі станції(Web Server 1), тобто середню кількість послуг, виконаних за одиницю часу.

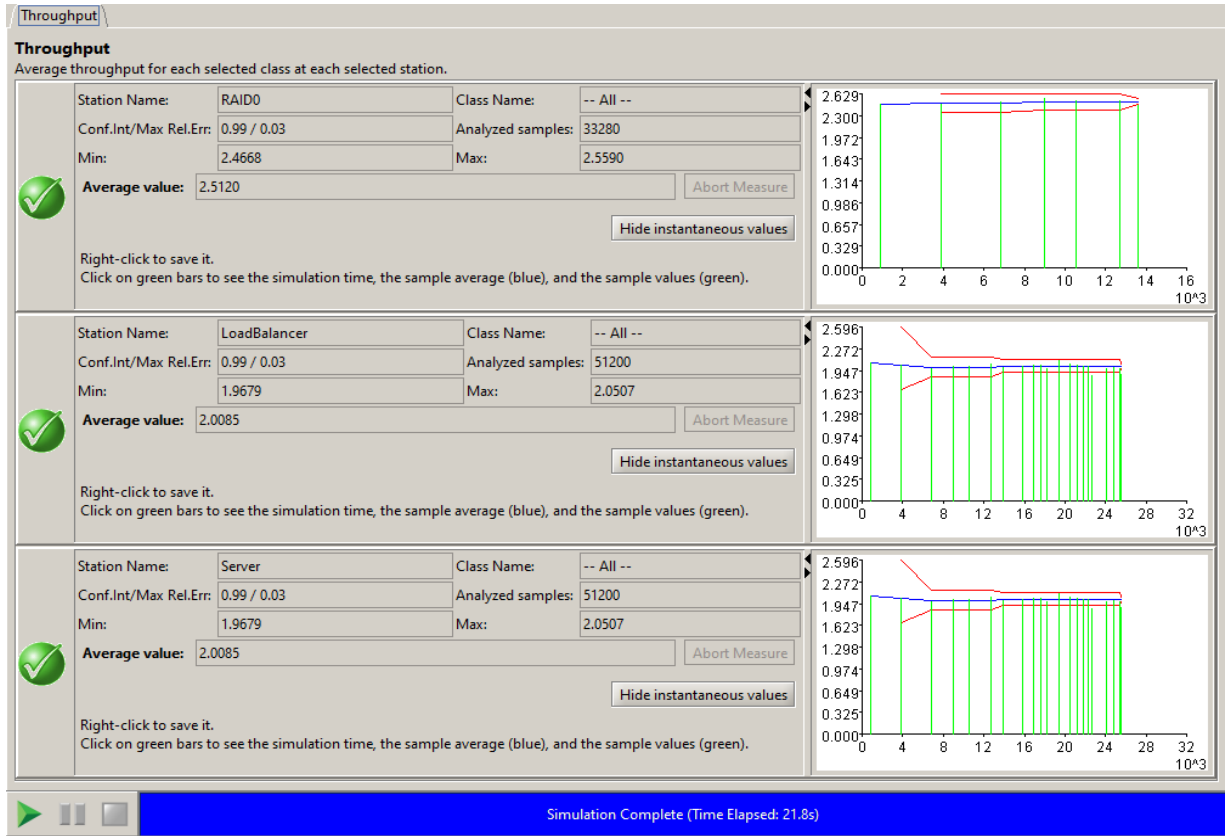


Рисунок 3.6 – Моделювання пропускної здатності окремих елементів системи

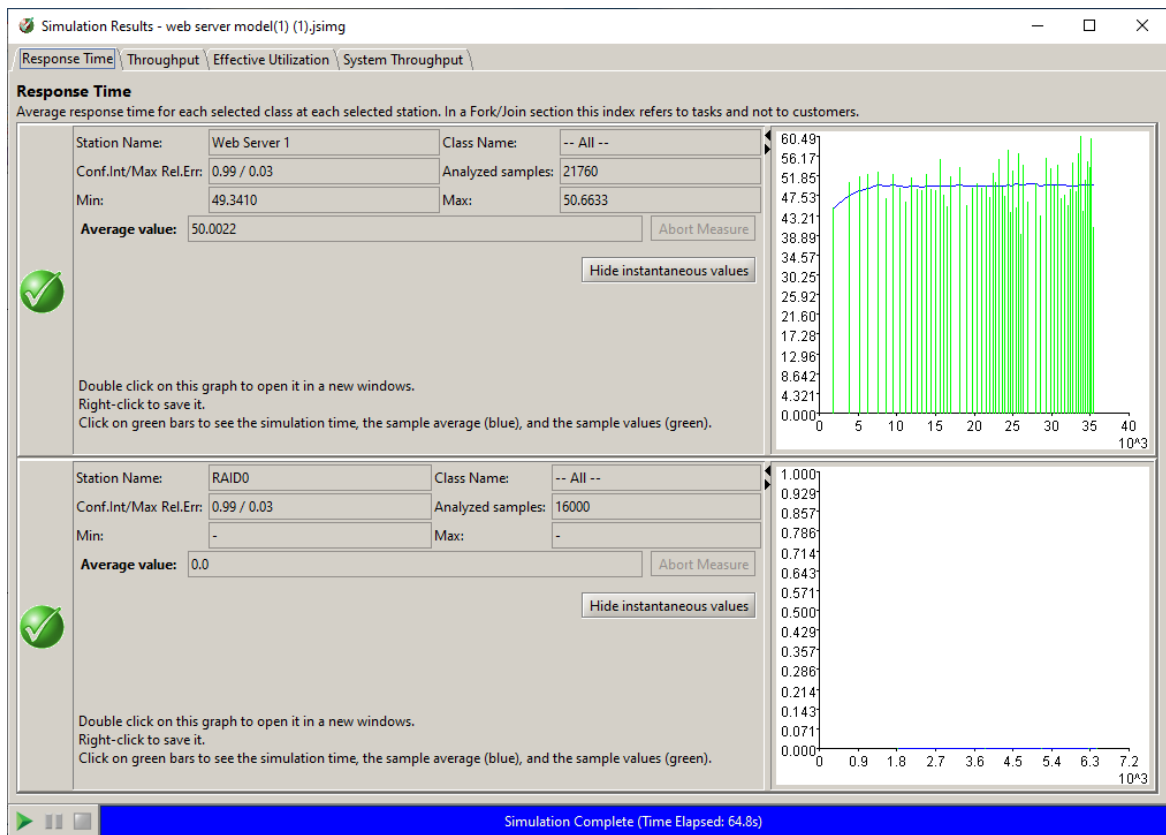


Рисунок 3.7 – Моделювання часу відгуку Web-сервера 1

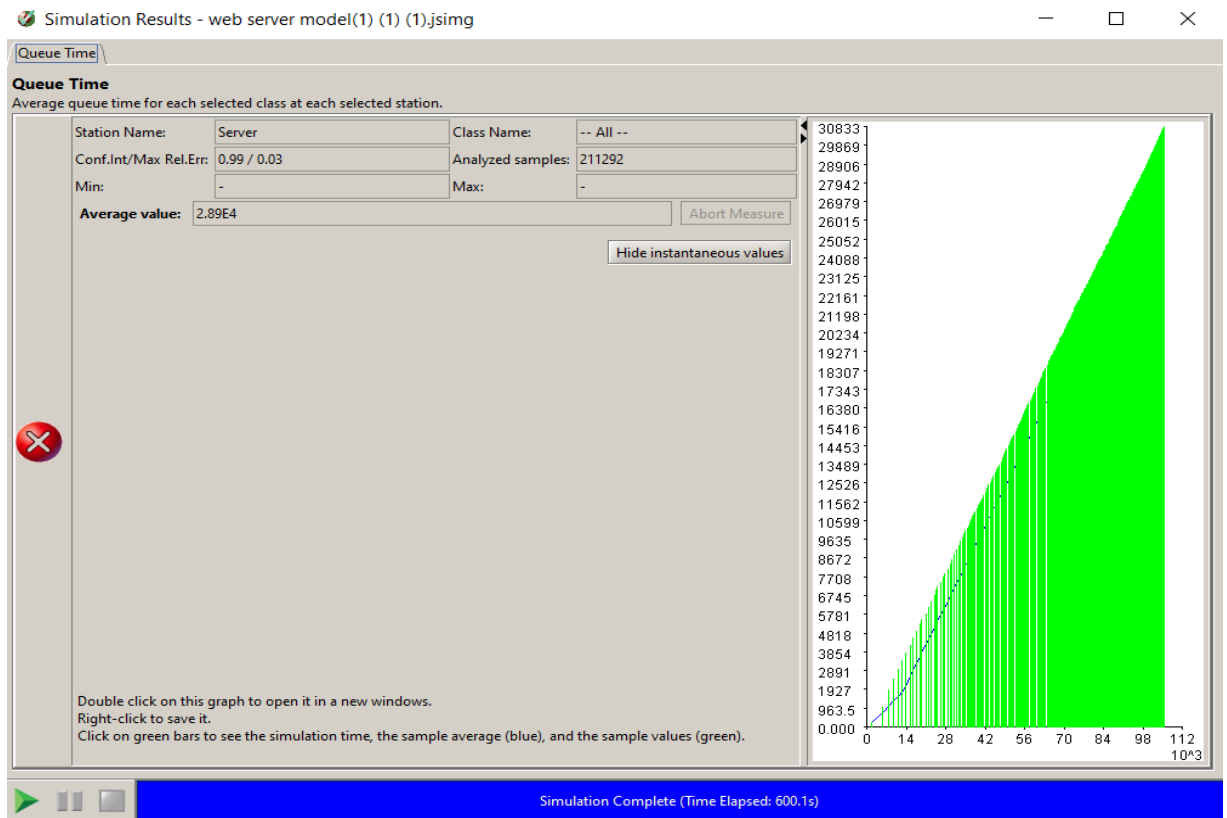


Рисунок 3.8 – Моделювання часу очікування в черзі системи

Як видно з моделювання(див. рис. 3.9), середній час, проведений клієнтами в черзі на станції неспинно збільшується. Цьому сприяло збільшення зростання часу відгуку системи.

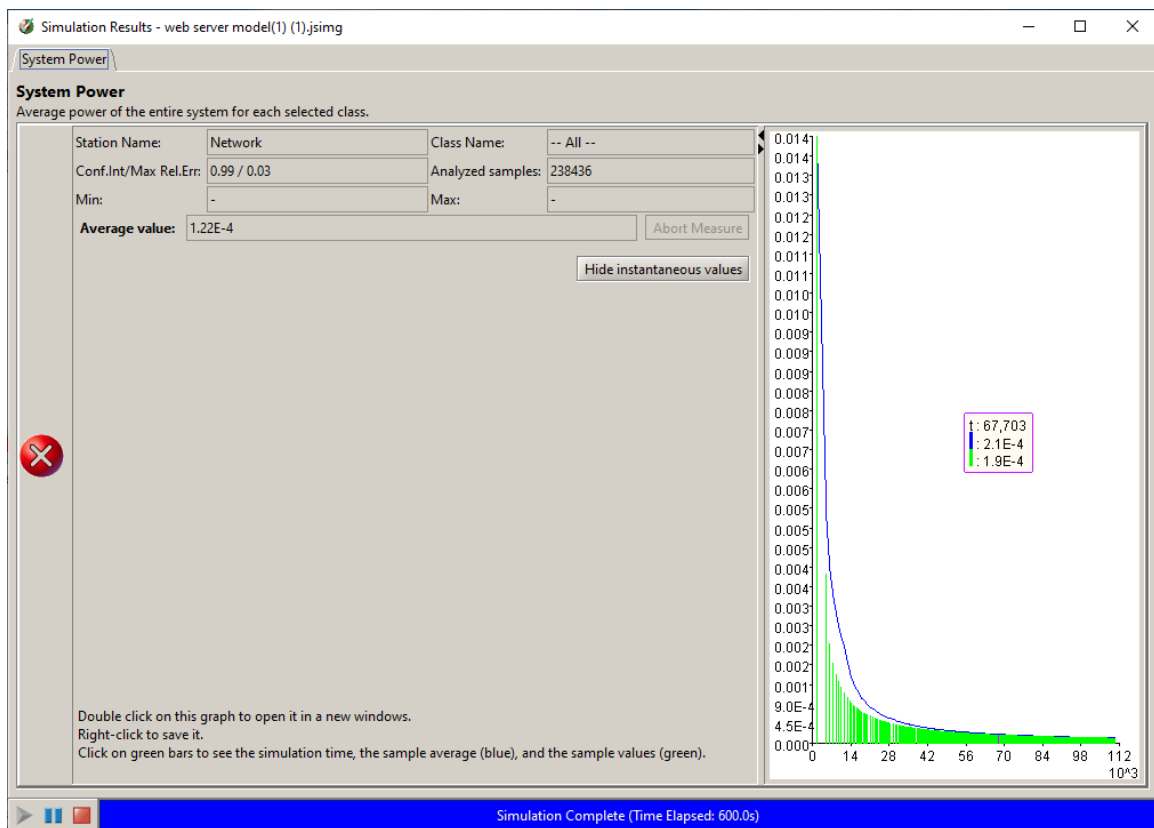


Рисунок 3.9 – Моделювання потужності системи

За результатами моделювання потужності системи оптимальною робочою точкою системи є середнє значення(Average value), що відповідає максимальній пропускній здатності системи.

Як видно з графіку(див. рис. 3.9), потужність системи є не стабільною, та постійно знижується.

Під час моделювання було проведені дослідження різних показників роботи системи та її елементів, зазначено, що система працює неефективно, і потребує оптимальних рішень для її подальшої експлуатації.

3.2 Аналіз та інтерпретація результатів дослідження

Аналіз даних моделювання показує, що час відгуку системи для Class0 в середньому становить $1,56 \cdot 10^4$ с, а для Class1 – $3,53 \cdot 10^4$ с. І, поступово цей час зростає.

Пропускна здатність системи становить 1 клієнт/с, тоді як пропускні здатності сервера становить 2,0 клієнта/с, масива RAID0 – 2,4 од/с.

Час відгуку Web-сервера 1 становить 59 с.

Потужність системи не є стійкою і постійно падає, за період моделювання вона становить $1,22 \cdot 10^{-4}$.

Система працює неефективно, оскільки за прийнятний час вона не може досягти стійкого стану. Це свідчить про те, що існують певні ризики, пов'язані з експлуатацією системи. Збільшення навантаження на систему може призвести до її розбалансування та виходу з ладу.

Основні проблеми в роботі системи пов'язані з дисципліною обслуговування черги, оскільки користувач, що потрапив до системи, якщо вона зайнята, не відкидається, а продовжує чекати звільнення системи. Це призводить до зростання часу відгуку системи. Вирішення цієї проблеми можливе в двох напрямках: відкидання певної кількості клієнтів, які чекають, або зменшення часу обслуговування клієнтів.

Перший спосіб потребує детального аналізу типів клієнтів та ризику втрати важливої інформації.

Другий спосіб потребує грошових вкладень, бо вирішити його можна за рахунок збільшення кількості серверів або використання додаткового обладнання.

ВИСНОВКИ

На основі наведених розділів та їх змісту можна зробити наступні висновки:

1) У роботі виконано аналіз основних засобів моделювання ІКС серед яких перевагу віддано пакету JMT.

2) Переваги JMT полягають у таких його характеристиках:

Універсальність, наочність опису моделі, подання результатів у графічному вигляді, можливість використання мереж Петрі для побудови моделей.

3) Для побудови імітаційних моделей ІКС та дослідження механізмів їхнього функціонування розглянуто основні елементи JMT та їхнє призначення для зазначеної задачі.

4) Побудова імітаційних моделей розглядається на прикладі двох серверної ІКС, що працює з дисковим простором за схемою RAID0 та описаним механізмом розподілу заявок.

5) Основні механізми функціонування та їхнє дослідження наведено в третьому розділі.

6) Усі вони зведені в загальну таблицю з описом виділених елементів і демонстрації графіків їхньої роботи.

7) Отримані в роботі результати можуть бути використані в процесі концептуального проектування інформаційно-керуючих систем для підприємств промисловості та транспорту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Serazzi G. Performance Engineering Learning Through Applications Using JMT. Milano, Pavia, Italy, 2023. 3-4 p. URL: https://link.springer.com/chapter/10.1007/978-3-031-36763-2_1
2. Sterman, J. D. Business Dynamics: Systems Thinking and Modeling for a Complex World. McGraw-Hill Education, 2002. 202-209 p.p. URL: https://www.researchgate.net/publication/44827001_Business_Dynamics_System_Thinking_and_Modeling_for_a_Complex_World
3. Fishwick, P. A. Simulation Model Design and Execution: Building Digital Worlds. Prentice Hall, PTR, Upper Saddle River, NJ, United States, 1995. 56-61 p.p. URL: <https://dl.acm.org/doi/10.5555/552048>
4. Zeigler, B. P., Praehofer, H., & Kim, T. G. Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press. San Diego, 2000. 134 p. URL: https://www.researchgate.net/publication/243547056_Theory_of_Modeling_and_Simulation_Integrating_Discrete_Event_and_Continuous_Complex_Dynamic_Systems
5. Ross, S. M. Introduction to Probability Models. In Encyclopedia of Statistical Sciences John Wiley & Sons. Los Angeles, California, 2006. 466-471 p.p. URL: https://fac.ksu.edu.sa/sites/default/files/introduction-to-probability-model-s.ross-math-cs.blog_.ir_.pdf
6. Law A. M., Kelton W.D., Simulation Modeling and Analysis. McGraw-Hill Education. Singapore, 1991. 3 p. URL: <https://fac.ksu.edu.sa/sites/default/files/index.pdf>
7. Law, A. M. Simulation Modeling and Analysis. Handbook in Operations Research and Management Science. 2007. 433-481 p.p. URL: <https://industri.fatek.unpatti.ac.id/wp-content/uploads/2019/03/108-Simulation-Modeling-and-Analysis-Averill-M.-Law-Edisi-5-2014.pdf>
8. Sterman, J. D. System Dynamics Modeling: Tools for Learning in a Complex World.

- California Management Review. 2001. 8-25 p.p. URL:
<https://journals.sagepub.com/doi/10.2307/41166098>
9. Bertoli M., Casale G., Serazzi G. The jmt simulator for performance evaluation of non-product-form queueing networks. Conference: Proceedings 40th Annual Simulation Symposium (ANSS-40 2007). Norfolk, Virginia, USA, 2007. 1-6 p.p URL:
https://www.researchgate.net/publication/220696789_The_JMT_Simulator_for_Performance_Evaluation_of_Non-Product-Form_Queueing_Networks
10. Java Modelling Tools – JMT. URL: <https://jmt.sourceforge.net/>
11. Casale G., Serazzi G. Java Modelling Tools: User manual. Politecnico di Milano, and Imperial College London. 2021. 235 p. URL:
https://jmt.sourceforge.net/Papers/JMT_users_Manual.pdf
12. Rabta B., Alp A., Reiner G. Queueing Networks Modeling Software for Manufacturing. 2007. URL:
https://www.researchgate.net/publication/226332891_Queueing_Networks_Modeling_Software_for_Manufacturing
13. Baccelli, F. Makowski A.M., Shwartz, A. The fork-join queue and related systems with synchronization constraints: stochastic ordering and computable bounds. Adv. Appl. Probab. 1989. 629–660 p.p. URL:
<https://www.jstor.org/stable/1427640>
14. Marin A., Rossi S., Sottana M. Biased Processor Sharing in Fork-Join Queues. In Proceedings of the International Conference on Quantitative Evaluation of Systems, Beijing, China, 2018. URL:
https://link.springer.com/chapter/10.1007%2F978-3-319-99154-2_17
15. Memon R.A., Li J.P., Ahmed J. Simulation Model for Blockchain Systems Using Queuing Theory. Electronics. 2019. 234 p. URL:
<https://doi.org/10.3390/electronics8020234>

16. Atayero A.A., Alatishe A.S., Iruemi J.O. Modeling and Simulation of a University LAN in OPNET Modeller Environment. Department of Electrical and Information Engineering, Covenant University, Nigeria. URL:
<https://core.ac.uk/download/pdf/12356407.pdf>
17. Andrei B. Multi-method modelling: AnyLogic. The AnyLogic Company, St Petersburg, Russia. 2014, 248-279 p.p. URL:
https://www.researchgate.net/publication/284696969_Multi-method_modelling_AnyLogic
18. Silvia D., Karol A., Pawel D. Simulation of a queuing system of a post office in Anylogic software. Communications - Scientific letters of the University of Zilina. 2020. URL:
<https://www.semanticscholar.org/reader/468f031f168a927cd7f12c53ab15c0297ac46015>
19. Gîlcă G. Ball and beam modeling using Matlab/Simulink. Lecturer PhD, “Constantin Brâncuși” University from Tîrgu-Jiu, Romania. 2019. 1-6 p.p. URL:
https://www.utgjiu.ro/rev_ing/pdf/2019-1/01_Gilca_Gheorghe BALL%20AND%20BEAM%20MODELING%20USING%20MATLAB%20SIMULINK.pdf
20. Zheng L., Hongji Y. Unlocking the Power of OPNET Modeler. Cambridge University Press, New York. 2012. 253 p. URL:
https://www.academia.edu/40530909/Book_UnlockingPowerOPNETModeler