

ДОДАТОК А

Графічний матеріал кваліфікаційної роботи

Харківський національний університет радіоелектроніки
Кафедра ЕОМ

Розробка структурної моделі інформаційної системи E-learning

КВАЛІФІКАЦІЙНА РОБОТА

Другий (магістерський)

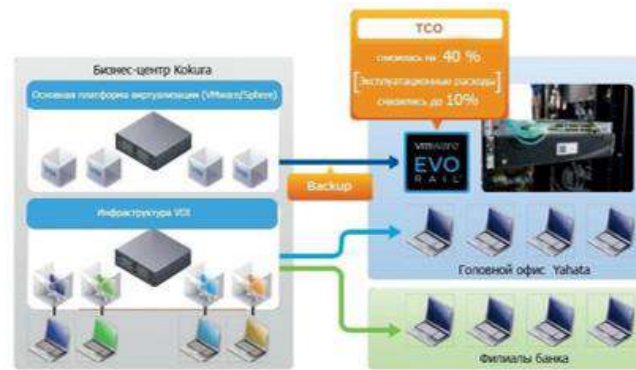
Автор
Коткова О.М.
студ. гр. СПзм 20-1

Керівник
Кучук Н.Г.
проф. каф. ЕОМ

Актуальність дослідження

- Розвиток інформаційних технологій стимулює персоналізацію освіти, інтеграцію педагогічних та інформаційних технологій, відкритість процесу освіти. Ці напрямки забезпечуються реалізацією концепції e-learning, котра використовуючи широкий набір застосунків, забезпечує навчання побудоване на використанні web-технологій.
- База e-learning – це електронні освітні ресурси.





Тенденції на ринку ІТ-технологій полягають у поступовому витисненні конвергентними і гіперконвергентними технологіями хмарних платформ. Для електронні освітні ресурси на гіперконвергентній інфраструктурі притаманне найшвидше розгортання системи, відсутня необхідність у складному проектуванні, при цьому для обслуговування серверу часом достатньо одного системного адміністратора. Це є найкращим рішенням для університетської e-learning в умовах дуже обмеженого бюджету.

3

Мета дослідження

Метою дослідження є розробка моделі функціонування системи e-learning на гіперконвергентній основі, котра враховує її особливості та дозволяє встановити інформаційні взаємозв'язки між складовими системи та провести аналіз базової мережі.

Задачі дослідження

Побудувати математичну модель (програмне рішення) інформаційної структури системи підтримки ЕОР університетської e-learning, а також:

- Ввести вихідні дані (з перевіркою обмежень).
- Сформувані вхідні дані.
- Для невеликих значень U , N , A , E , D сформувані граф інформаційної структури.

4

Необхідність в моделюванні інформаційної інфраструктури

Проблеми аналізу структури гіперконвергентної базової мережі:

- відсутній єдиний підхід до формування структури;
- залежність структури мережі від параметрів електронних освітніх ресурсів;
- відсутні відпрацьовані методи формального опису структури мережі, які можна використати при проведенні розрахунків

5

Недоліки ММО:

1. ММО вимагає розрізнення потоків запитів і даних;
2. завантаження кожного каналу визначається інтенсивністю потоків даних усіх типів, а не тільки потоків запитів;
3. застосунок може багаторазово взаємодіяти з іншими застосунками на інших вузлах мережі, що ускладнює процес аналізу;
4. передача по каналах зв'язку транзитних потоків;
5. безліч непотрібних результатів аналізу ММО.

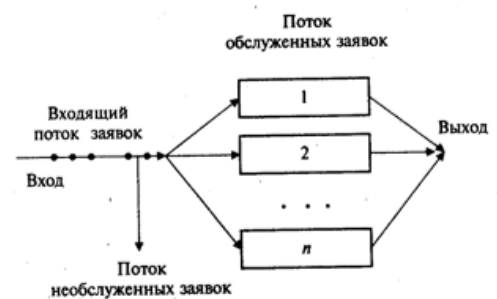
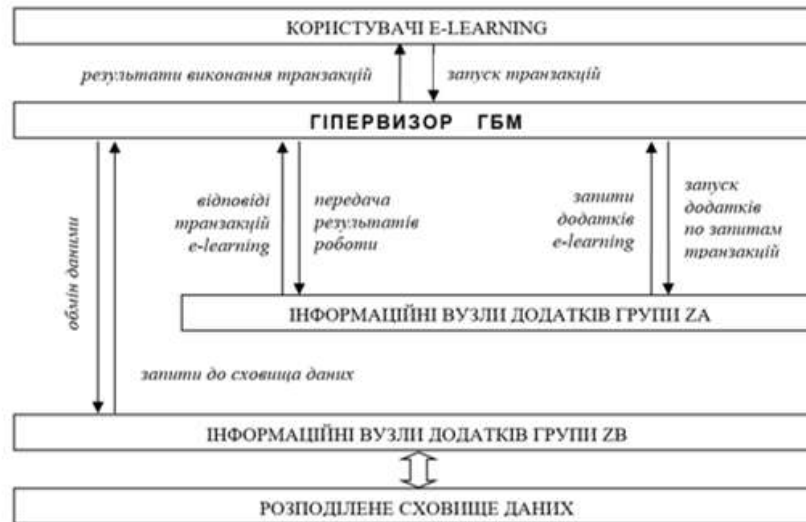


Рис.1. Схема СМО

6

Розроблена модель основних інформаційних взаємозв'язків у середовищі e-learning



7

МАТЕМАТИЧНЕ УЯВЛЕННЯ МОДЕЛІ

M_U – множина користувачів,
 M_N – множина вузлів,
 M_A – множина додатків,
 M_E – множина транзакцій,
 M_D – множина блоків даних.

$E_e = \langle \overline{a_e}, \overline{d_e}, \overline{u_e}, \overline{w_e} \rangle$ – кортеж, що описує параметри кожної транзакції із використаними додатками, блоками даних, користувачами, що використовують та порядком запуску.

8

ЛОГІЧНЕ УЯВЛЕННЯ МОДЕЛІ



10

$A_{ae} = \langle \lambda_{ae}, \beta_{ae} \rangle$ – параметри кожного додатку відповідної транзакції.

Також маємо булеві матриці: G – розміщення додатків по вузлам мережі, H – розміщення користувачів по вузлам мережі, S – розміщення блоків даних по вузлам мережі.

Відповідно до цього маємо кортеж, що описує розроблену модель:

$$\mathfrak{J} = \langle M_U, M_N, M_E, M_D, \{E_e\}, \{A_{ae}\}, G, H, S \rangle$$

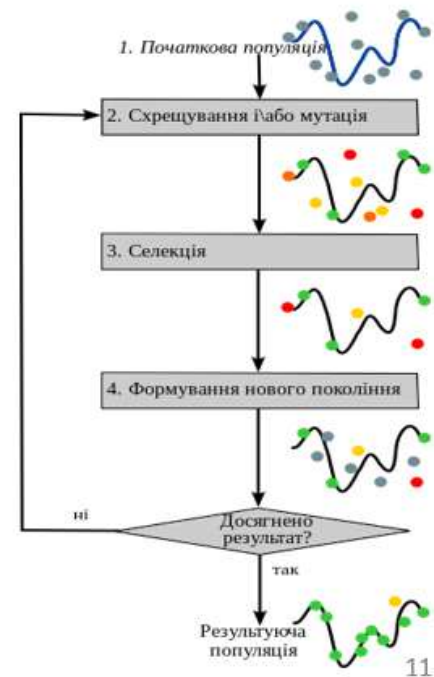
9

ГЕНЕТИЧНИЙ АЛГОРИТМ

Головним критерієм для оцінки якості інформаційної структури є збалансованість навантаження на вузли.

Переваги генетичних алгоритмів:

- концептуальна простота;
- дозволяє простіше враховувати апріорні знання;
- можливість варіювання часом пошуку раціонального рішення за допомогою зміни вимог до фітнес-функції.



11

Переваги над традиційними методами дискретної оптимізації пошуку оптимальних рішень (метод Дейкстри, метод гілок та меж, динамічне програмування):

- алгоритм працює тільки із кодами, в процесі роботи код розглядається тільки як бітовий рядок (без врахування змістового сенсу);
- на кожному кроці обробляється лише декілька точок простору кодів;
- в процесі роботи алгоритм не потребує ніякої додаткової інформації, це сприяє підвищенню швидкості алгоритму;
- для формування нових точок простору використовується одночасно ймовірнісні та детерміновані методи, що дає значно більший ефект, ніж окреме використання.

12

Переваги над нейронними мережами:

- не потрібно формувати навчаючу множину
- процес початкової підготовки набагато простіший.

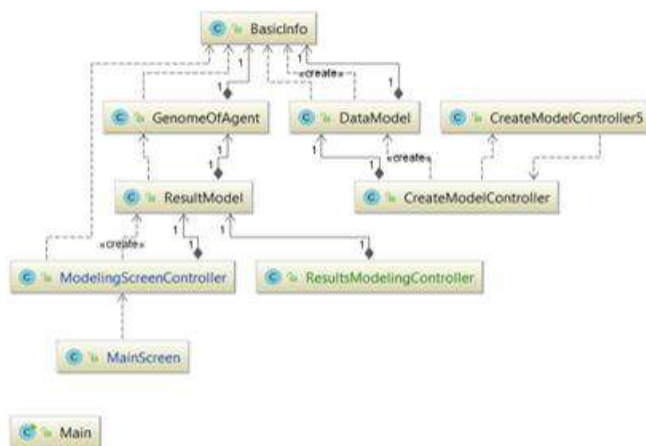
Головна перевага – дозволяє знайти прийняття рішення за менший час, ніж інші алгоритми.

Недоліки:

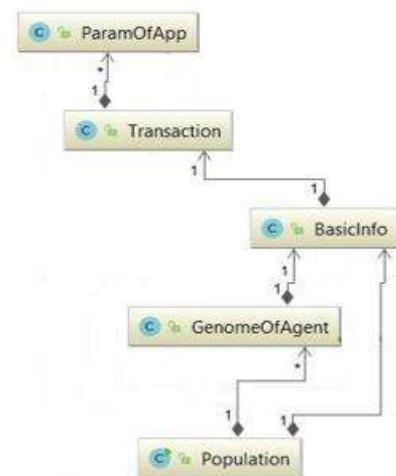
- отримуємо раціональне, а не оптимальне рішення
- неможливо отримати статистичну оцінку отриманого рішення.

13

Діаграма класів



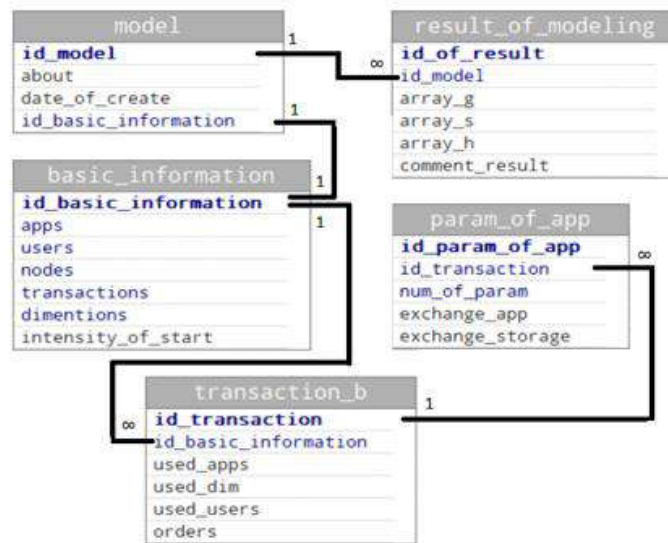
Графічний інтерфейс



Ядро генетичного алгоритму

14

Схема бази даних



15

РЕЗУЛЬТАТИ РОБОТИ

БАЗОВАЯ ИНФОРМАЦИЯ ОТМЕНА

КОЛИЧЕСТВО ПРИЛОЖЕНИЙ

КОЛИЧЕСТВО УЗЛОВ

КОЛИЧЕСТВО ПОЛЬЗОВАТЕЛЕЙ

КОЛИЧЕСТВО ТРАНЗАКЦИЙ

КОЛИЧЕСТВО БЛОКОВ ДАННЫХ

ДАЛЕЕ



ИНТЕНСИВНОСТИ ЗАПУСКА ТРАНЗАКЦИЙ ОТМЕНА

---> ТРАНЗАКЦИИ --->

Пользователи	1	2	3	4	5
1	25	0	0	0	0
2	0	10	10	0	0
3	10	0	0	13.4	0
4	7	0	0	0	25
5	0	13	0	0	9
6	0	0	7	0	0

ДАЛЕЕ

СОХРАНИТЬ ВИЗУАЛИЗИРОВАТЬ ОТМЕНА

G:

```

0000000100
1000000000
0000100000
0100000000
0010000000
0001000000
0000000010
0000001000
0000000000
S:
0001000000
0000100000
0010000000
0000000100
0000000001
0000000000
    
```

16

ВИСНОВКИ

- Результатом виконання дипломної роботи стала програма, що здатна через графічний інтерфейс приймати від користувача вхідні дані, що характеризують склад системи e-learning із власними додатками, блоками даних і транзакціями що описують їх взаємодію та інформацію про склад гіперконвергентної мережі і на виході отримувати розташування користувачів, блоків даних і додатків по вузлам мережі, що являється оптимальною в умовах даної інтенсивності запуску транзакцій користувачами. Дана програма є досягненням поставленої мети, та реалізує усі поставленні задачі.
- Розроблена програма дозволить підвищити ефективність використання базової гіперконвергентної мережі, а, отже, і підвищити якість функціонування системи e-learning в цілому, що являється необхідною складовою створення такої системи в умовах обмеженого бюджету університету.

ДОДАТОК Б
Детальна структура класів

C ParamOfApp		
f	exchangeBetweenApp	int[]
f	exchangeBetweenStorage	int[]

Рисунок Б.1 – Вміст класу ParamOfApp.java

C Transaction		
	a	int[]
	d	int[]
	u	int[]
	w	int[][]
	paramOfApps	ParamOfApp[]

Рисунок Б.2 – Вміст класу Transaction.java

C GenomeOfAgent		
	basicInfo	BasicInfo
	fitnessVal	double
	isChanged	boolean
	mG	int[][]
	mH	int[][]
	mS	int[][]
	setBasicInfo(BasicInfo)	void
	setmG(int[][])	void
	setmH(int[][])	void
	setmS(int[][])	void
	getmG()	int[][]
	getmS()	int[][]
	getmH()	int[][]
	getFitnessVal()	double
	mutate(int)	GenomeOfAgent
	crossing(GenomeOfAgent)	GenomeOfAgent
	workOfApp()	int[][]
	workOfStorage()	int[][]
	dataInNetwork()	int[][]
	loadOfNodesByTransactions()	double[][]
	loadOfNetwork()	double[][]
	averageLoad()	double
	fitness()	double
	intensityOfStartApp()	double[][]
	loadNodesStorage()	double[][]
	intensityStorage()	double[][]

Рисунок Б.3 – Вміст класу GenomeOfAgent.java

Population		
f	BY_TARGET_VALUE	int
f	BY_NUM_OF_GENERATION	int
f	MAX_COUNT_OF_MUTATE	int
f	agents	ArrayList<GenomeOfAgent>
f	basicInfo	BasicInfo
f	countOfAgents	int
f	targetValue	int
f	countForMutate	int
f	countForCrossing	int
m	doEvolutionProcess(int)	void
m	generateNewPopulation()	void
m	sortByFitness()	void
m	getBestInx()	int
m	generateRandomPopulation(int)	ArrayList<GenomeOfAgent>
m	generateArr(int, int)	int[][]
m	getAgents()	ArrayList<GenomeOfAgent>
m	main(String[])	void

Рисунок Б.4 – Вміст класу Population.java

DBCore		
f	DB_NAME	String
f	connection	Connection
f	statement	Statement
m	createDB()	void
m	closeConnection()	void
m	getParamOfAppByTransactionID(int, int)	ParamOfApp[]
m	getTransactionByBasicInfoID(int, int, int)	Transaction[]
m	getBasicInfoByID(int)	BasicInfo
m	getGenomeOfAgentByModelID(int, int)	GenomeOfAgent
m	saveToDBParamOfApp(ParamOfApp, int, int)	void
m	saveToDBTransaction(Transaction, int)	void
m	saveToDBResult(int, GenomeOfAgent, String)	void
m	deleteResultOfModelingByID(int)	void
m	deleteResultOfModelingByModelID(int)	void
m	deleteModelByID(int)	void
m	saveToDBModel(BasicInfo, String, String)	void
m	getSimpleModels()	ArrayList<SimpleModel>
m	main(String[])	void

Рисунок Б.5 – Вміст класу DBCore.java

DataModel		
f	aboutModel	String
f	dateCreate	String
f	basicInfo	BasicInfo
J	getBasicInfo()	BasicInfo
J	getAboutModel()	String
J	getDateCreate()	String
J	setAboutModel(String)	void
J	setDateCreate(String)	void
J	setBasicInfo(BasicInfo)	void

Рисунок Б.6 – Вміст класу DataModel.java

ResultModel		
f	idModel	int
f	commentResult	String
f	agent	GenomeOfAgent
m	setAgent(GenomeOfAgent)	void
m	setCommentResult(String)	void
m	setIdModel(int)	void
m	getAgent()	GenomeOfAgent
m	getIdModel()	int
m	getCommentResult()	String

Рисунок Б.7 – Вміст класу ResultModel.java

ResultsModelingController		
f	resultModel	ResultModel
f	mResultsScroll	ScrollPane
m	initialize()	void
m	doCancel(ActionEvent)	void
m	doSave(ActionEvent)	void
m	doVisualization(ActionEvent)	void

Рисунок Б.8 – Вміст класу ResultsModelingController.java

C ModelingScreenController		
f	models	ObservableList<String>
f	modelsArr	ArrayList<SimpleModel>
f	resultModel	ResultModel
f	mCountAgents	TextField
f	mCountMutation	TextField
f	mCountCross	TextField
f	mCountIterFitness	TextField
f	mBasicInfoCombo	ComboBox
f	mButtonByCount	RadioButton
f	mButtonToValue	RadioButton
m	initialize()	void
m	doStartModeling(ActionEvent)	void
m	doCancelModeling(ActionEvent)	void

Рисунок Б.9 – Вміст класу ModelingScreenController.java

C BinaryObjectConverter		
m	byteToArr1DI(byte[])	int[]
m	byteToArr2DI(byte[])	int[][]
m	byteToArr2DD(byte[])	double[][]
m	arr1DIToByte(int[])	byte[]
m	arr2DIToByte(int[][])	byte[]
m	arr2DDToByte(double[][])	byte[]

Рисунок Б.10 – Вміст класу BinaryObjectConverter.java

C BasicInfo		
f	U	int
f	N	int
f	A	int
f	E	int
f	D	int
f	intensityOfRun	double[][]
f	transactions	Transaction[]
m	calcLoadOfTransaction()	double[]
m	calcLoadOfApp()	double[]

Рисунок Б.11 – Вміст класу BasicInfo.java

C CreateModelController		
f	dataModel	DataModel
f	isNotMade	boolean
f	intensityText	TextField[]
f	usedApp	CheckBox[]
f	usedDim	CheckBox[]
f	usedUsers	CheckBox[]
f	usingOrder	CheckBox[]
f	numOfTransaction	int
f	isFilledTrs	boolean[]
f	paramOfApps	ParamOfApp[]
f	textArea	TextArea
f	datePicker	DatePicker
f	countAppTextField	TextField
f	countNodeTextField	TextField
f	countUsersTextField	TextField
f	countTransactionTextField	TextField
f	countDimTextField	TextField
f	scrollIntensity	ScrollPane
f	myIndexation	Label
f	usedAppScrollPane	ScrollPane
f	usedDimScrollPane	ScrollPane
f	usedUsersScrollPane	ScrollPane
f	orderRunScrollPane	ScrollPane
m	doCancel(ActionEvent)	void
m	doNextOne(ActionEvent)	void
m	doNextTwo(ActionEvent)	void
m	doNextThree(ActionEvent)	void
m	doActivateFields(MouseEvent)	void
m	doActivateFieldsByTrs(MouseEvent)	void
m	doNextTrs(MouseEvent)	void
m	doPerviousTrs(MouseEvent)	void
m	doFillParamApp(MouseEvent)	void
m	doNextPerious(boolean)	void
m	doSave(ActionEvent)	void

Рисунок Б.12 – Вміст класу CreateModelController.java

C CreateModelController5		
f	isFiled	boolean
f	usedAppInt	int[]
f	usedDimInt	int[]
f	paramOfApps	ParamOfApp[]
f	inxOfTrs	int
f	usedAppsTextFields	TextField[]
f	usedDimTextFields	TextField[]
f	inxUsedApp	int
f	isNotActivated	boolean
f	bAppScrollPane	ScrollPane
f	bDimScrollPane	ScrollPane
f	myIndexation	Label
f	myTrsIndexation	Label
m	doActivateFields(MouseEvent)	void
m	doNextApp(MouseEvent)	void
m	doPerviousApp(MouseEvent)	void
m	doCancel(ActionEvent)	void
m	doNextTask(ActionEvent)	void
m	stepNext()	int
m	stepPrvs()	int
m	doStep(boolean)	void

Рисунок Б.13 – Вміст класу CreateModelController5.java

C MainScreen		
m	initialize()	void
m	doExitButton(ActionEvent)	void
m	doStartModeling(MouseEvent)	void
m	doCreateModel(MouseEvent)	void
m	doRollupButton(ActionEvent)	void
m	doMoreButton(ActionEvent)	void
m	doMovingByMouse(MouseEvent)	void

Рисунок Б.14 – Вміст класу MainScreen.java

C Main		
m	start(Stage)	void
m	main(String[])	void

Рисунок Б.15 – Вміст класу Main.java