

УДК 004.75

DOI: 10.25140/2411-5363-2020-4(22)-131-143

Володимир Русінов, Богдан Іваніщев, Андрій Антонюк,
Артем Волокита, Георгій Луцький

СПОСОБИ СИНТЕЗУ ТОПОЛОГІЧНИХ ОРГАНІЗАЦІЙ НА ОСНОВІ КОДОВИХ ПЕРЕТВОРЕНЬ ДЕБРУЙНА

Актуальність теми дослідження. На сьогодні такі топологічні організації, як дерева (tree), жирні дерева (fat tree), 3D-mesh, 6D-tor, DragonFly використовуються як спосіб мережевої організації в центрах обробки даних. Усі ці топології мають відмовостійкі властивості внаслідок наявності декількох альтернативних шляхів маршрутизації. Тому актуальною є задача розробки нових топологій, які будуть мати багато збалансованих маршрутів (one cost path). Одним із кращих варіантів для досліджень покращення топології дерева є комбінація з графами де Бруїна в різних системах числення, що дозволяє варіювати ступенем вершин, діаметром та кількістю альтернативних шляхів для маршрутизації. У статті пропонується використати новий спосіб, який дозволяє отримати задані характеристики при об'єднанні топологій на основі дерев з графом на основі кодових перетворень де Бруїна.

Постановка проблеми. При створенні високопродуктивних обчислювальних систем важливою є завчасна розробка їх топологічних організацій, яка дозволяє вчасно передбачити характеристики системи.

Аналіз останніх досліджень і публікацій. На цей час існує низка робіт, присвячених топологіям, що засновані на послідовності де Бруїна. Існує детальний огляд синтезу гібридних топологій на основі графів де Бруїна та гіперкуба.

Виділення недосліджених частин загальної проблеми. Не розглянутою на цей час залишається можливість використання гібридних топологій на основі дебруїнівських послідовностей та різних типів дерев.

Постановка завдання. Завданнями є розробка способу синтезу нових топологій на основі дерев і дебруїнівських послідовностей та аналізу характеристик синтезованих топологічних організацій для різних рівнів відмовостійкості.

Виклад основного матеріалу. Описано синтез n -яти топологій на основі дерев і дебруїнівських послідовностей. Проведено порівняльний аналіз усіх синтезованих топологій.

Висновки відповідно до статті. Виконано аналіз характеристик, виділено основні переваги та недоліки запропонованих топологічних структур, висунуто пропозиції щодо їх покращення.

Ключові слова: граф де Бруїна; топологія; відмовостійкість.

Рис.: 13. Табл.: 9. Бібл.: 13.

Актуальність теми дослідження. На сьогодні дерева (tree) та жирні дерева (fat tree) широко використовуються як спосіб мережевої організації в центрах обробки даних. Також в сучасних суперкомп'ютерах використовуються такі топологічні організації, як 3D-mesh [1], DragonFly [2], 6D-tor [3]. Усі ці топології мають відмовостійкі властивості внаслідок наявності декількох альтернативних шляхів маршрутизації. Тому актуальною є задача розробки нових топологій, які будуть мати багато збалансованих маршрутів (one cost patch) [4]. Одним з варіантів для досліджень покращення топології дерева є комбінація з графами де Бруїна в різних системах числення. Використання різних систем числення дозволяє варіювати ступенем вершин, діаметром та кількістю альтернативних шляхів для маршрутизації. У статті пропонується використати новий спосіб, який дозволяє отримати задані характеристики при об'єднанні топологій на основі дерев з графом на основі кодових перетворень де Бруїна.

Постановка проблеми. При створенні високопродуктивної обчислювальної системи, важливою частиною роботи є розробка топологічної організації. Цей крок дозволяє вчасно передбачити характеристики системи, не виконуючи повної моделі системи, що дозволяє правильно підібрати обладнання та розробити алгоритми маршрутизації. Як відомо, більшість кластерів використовує центральні процесори разом з декількома графічними прискорювачами для розв'язання обчислювальних задач. Зазвичай, значна частина ліній PCI express вже використовується, і тому важливо передчасно кількість зв'язків для кожного вузла, що буде впливати на чергу повідомлень, яка буде утворюватися.

Аналіз останніх досліджень і публікацій. На цей час існує низка робіт, присвячених топологіям, що засновані на послідовності де Бруїна [5-7]. Існує детальний огляд синтезу гібридних топологій на основі графів де Бруїна та гіперкуба. Також цікаво зазначити, що збільшується кількість наукових статей, які використовують графи де Бруїна для синтезу ДНК і геномів [8].

Виділення недосліджених частин загальної проблеми. Однією з широко розповсюджених модифікацій дерева є товсте дерево, показники якого дозволяють виконувати масштабування відмовостійких топологій. Проте раніше не було розглянуто можливість використання гібридних топологій на основі дебруйнівських послідовностей та різних типів дерев.

Мета статті. Метою дослідження є розробка способу синтезу нових топологій на основі дерев і дебруйнівських послідовностей, та аналізу характеристик синтезованих топологічних організацій для різних рівнів відмовостійкості.

Виклад основного матеріалу. Більшість топологій, які застосовуються в суперкомп'ютерах, насправді мають корені з простих топологічних організацій, наприклад решітки (mesh). Отже, здійснити перехід від простих до складних топологій можна декількома шляхами:

Ієрархічний підхід. На основі кластера, який містить невелику кількість поєднаних між собою вузлів, формується ієрархічна топологія, в якій зв'язки між кластерами також відповідають одному заданому правилу.

Декартовий добуток топологій. Дозволяє отримати нові топології з поєднанням властивостей обох початкових топологій [5].

Об'єднання топологій. За допомогою додаткових зв'язків, можна зберегти переваги деякої топології та розширити її функціонал коштом зв'язків, що притаманні іншій топології.

До основних показників топологій будемо відносити кількість N вузлів топології, ступінь S топології, діаметр D топології, середній діаметр D топології, топологічний трафік Q , вартість C .

Як початкові базові досліджувані топологічні організації використані дерева та графи де Бруйна. Розглянемо звичайне бінарне дерево. Масштабування такої топології здійснюється наступним чином: є коренева вершина, на наступному кроці до неї додаються дві нові вершини, які називаються ярусом. На кожному кроці масштабування до останнього ярусу, до кожної з вершин додається по дві вершини, утворюючи новий ярус (рис. 1, табл. 1).

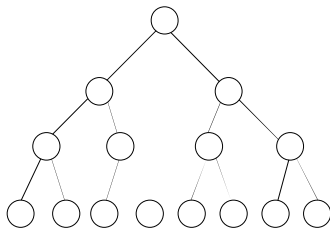


Рис. 1. Топологія бінарне дерево

Таблиця 1

Характеристики бінарного дерева на r -ому кроці

Параметр (r – ранг)	Кількість вершин	Ступінь	Діаметр	Ціна
Рівняння зросту	2^r-1	3	$2*(r-1)$	$3*(2^r-1)* *2*(r-1)$

Проте в дерева залишаються проблеми, які потребують вирішення. Наприклад, швидке зростання діаметру та топологічного трафіку. Одним із варіантів вирішення цієї проблеми є зупинка масштабування дерева з одночасним додаванням нових зв'язків, що реалізовано в топології товстого дерева. Інший варіант, який розглядається в даній роботі, - це синтезування гібридної топології на основі дерева та дебруйнівських зв'язків.

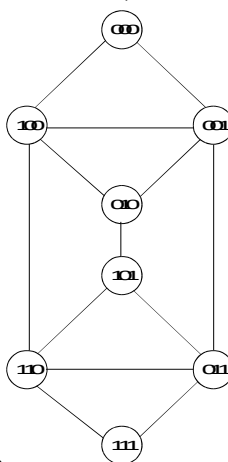
Розглянемо граф де Бруйна, який створюється за допомогою зсувів вліво. При формуванні нової послідовності, старша цифра витискається (зникає), молодша цифра після зсуву отримує значення 0 або 1. Отримані таким чином вузли мають зв'язок із вузлом, від номера якого зроблено зсув. Наприклад, візьмемо (табл. 2) де наведено формування послідовності де Бруйна для трізначних чисел.

Таблиця 2

Таблиця формування дебруйнівських послідовностей

000		001		010		011	
001	000	010	011	100	101	110	111
100		101		110		111	
000	001	010	011	100	101	110	111

Таким чином утворюється послідовність де Бруйна. Для масштабування графу, додається ще один розряд, що збільшує кількість вузлів в два рази. Потрібно зазначити, що граф можна повторно використати для наступних кроків масштабування. При цьому потрібно продублювати граф два рази, і додати нові зв'язки, тому що при збільшенні кількості розрядів, дебруйнівські зсуви будуть генерувати нові значення, які розширяють попередній етап масштабування (рис. 2, табл. 3).



0

Рис. 2. Топологія де Бруйна

Таблиця 3

Характеристики топології де Бруйна на r-ному кроці

Параметр (r – ранг)	К-сть вершин	Ступінь	Діаметр	Ціна
Рівняння зросту	2^r	4	r	$4 * 2^{r * r}$

Синтез топології. Перша запропонована топологія: **Кластери з двійкових дерев, що ієрархічно об'єднані за принципом де Бруйна.** Пропонується взяти двійкове дерево як кластер (групу вузлів) та об'єднати з іншими кластерами за логікою, що описується дебруйнівськими зсувами.

Розглянемо більш детально, як формуються зв'язки на основі дебруйнівських зсувів, на прикладі двійкового числа. Для формування зв'язку потрібно зробити зсув числа A вліво, та на місці останньої цифри записати 0 або 1. Таким чином, формуються два зв'язки, при цьому старша цифра, яка залишається поза розрядною сіткою, виштовхується і не враховується. Число A можна представити в наступному вигляді:

$$A = a_i * 2^i + a_{i-1} * 2^{i-1} + \dots + a_1 * 2 + a_0$$

Для утворення зв'язків, необхідно зробити зсув вліво, таким чином утворюється число A_s :

$$A_s = a_{i-1} * 2^i + a_{i-2} * 2^{i-1} + \dots + a_0 * 2 + E,$$

де E – приймає два значення 0 і 1.

Отже, числа з якими буде встановлений зв'язок A_{S0} і A_{S1} описуються такими формулами:

$$A_{S0} = a_{i-1} * 2^i + a_{i-2} * 2^{i-1} + \dots + a_0 * 2 + 0$$

$$A_{S1} = a_{i-1} * 2^i + a_{i-2} * 2^{i-1} + \dots + a_0 * 2 + 1$$

Варто зазначити, що існує два числа B_0 та B_1 , які матимуть зв'язок з числом A :

$$B_0 = 0 * 2^i + a_i * 2^{i-1} + \dots + a_2 * 2 + a_1$$

$$B_1 = 1 * 2^i + a_i * 2^{i-1} + \dots + a_2 * 2 + a_1$$

Розглянемо приклад для 4 кластерів (груп), де кожен кластер є бінарним деревом з 7 вершинами (рис. 3, табл. 4).

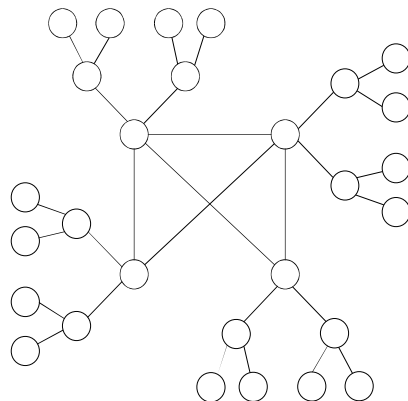


Рис. 3. Кластери з двійкових дерев, що ієрархічно об'єднані за принципом ДеБруїна

Таблиця 4

Характеристики кластерів двійкових дерев, що ієрархічно об'єднані за принципом ДеБруїна

К-сть вузлів	Діаметр	Ступінь	Середній діаметр	Ціна	Топологічний графік	SD
14	5	3	3,131868	210	2,087912	15
28	6	5	3,637566	840	1,445026	30
56	7	6	4,258442	2352	1,419481	42
112	8	6	4,852156	5376	1,617385	48
224	9	6	5,5217	12096	1,840567	54
448	10	6	6,256492	26880	2,085497	60
896	11	6	7,039867	59136	2,346623	66
1792	12	6	7,866417	129024	2,662214	72
3584	13	6	8,730765	279552	2,910255	78

Наступна запропонована топологія: **Двійкове дерево з горизонтальними дебруїнівськими зв'язками**. На кожному горизонтальному ярусі дерева, формуються зв'язки між вузлами на основі дебруїнівських зсувів.

Отже, на кожному наступному кроці масштабування до дерева додається $2^r - 1$ вершин, що також дорівнює кількості вершин в графі де Бруїна на r розрядах. Таким чином, ми можемо застосувати дебруїнівські зсуви на кожному ярусі дерева, для поліпшення базових характеристик дерева. При цьому рекомендуємо використовувати подвійну нумерацію, щоб органічно поєднати пошук по дереву і маршрутизацію за де Бруїном. Ярус визначається за допомогою номеру вузла в дереві. Припустимо, що нас цікавить, на якому ярусі знаходиться вузол p , для цього знаходимо значення $\lceil \log_2(p) \rceil = m$, яке відповідатиме ярусу, на якому знаходиться вершина. Знаючи це значення, можна визначити усі зв'язки між вершинами на ярусі вже з нумерацією де Бруїна. Таким чином, вузол має подвійний номер, що пишеться через крапку. Наприклад, 3.7, де 3 – це рівень дерева, а 7 – номер у топології де Бруїна (рис. 4, табл. 5).

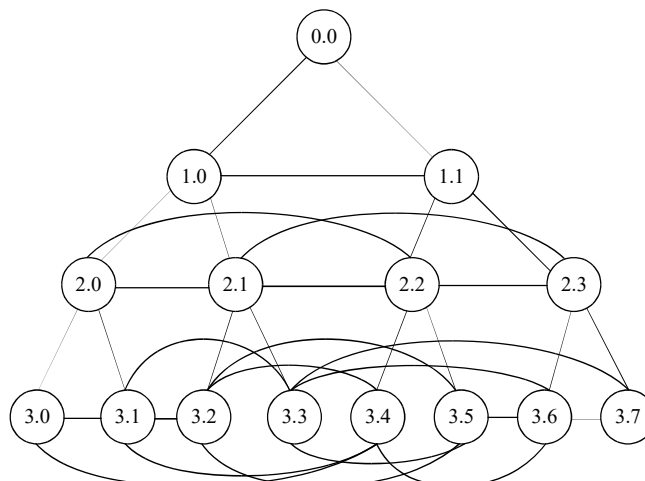


Рис. 4. Двійкове дерево з горизонтальними дебруйнівськими зв'язками

Таблиця 5

Характеристики двійкового дерева з горизонтальними дебруйнівськими зв'язками

К-сть вузлів	Діаметр	Ступінь	Середній діаметр	Ціна	Топологічний графік	SD
3	1	2	1	6	1	8
7	3	4	1,80952	84	0,90476	12
15	5	4	2,94286	300	1,47143	20
31	7	4	4,34409	868	2,17204	28
63	9	4	5,94031	2268	2,97465	36
127	11	4	7,69929	5588	3,84964	44
255	13	4	9,54618	13260	4,77309	52
511	15	4	11,45497	30660	5,72749	60
1023	17	4	13,40185	69564	6,70092	68
2047	19	4	15,37146	155572	7,68573	76
4095	21	4	17,35434	343980	8,67717	84

Третя запропонована топологія: **Двійкові дерева, пов'язані між собою трійковими дебруйнівськими зв'язками.** Кластери формуються на основі дерев на обраному кроці масштабування та об'єднуються за допомогою трійкових зсувів де Бруйна.

Цей спосіб схожий до першого розглянутого, проте трійкові дебруйнівські зсуви дещо по-іншому формуються. Як і в першому прикладі, як кластер обрано бінарне дерево з 7 вершинами. Для формування зв'язку потрібно зробити зсув числа А вліво, та на місці молодшої цифри записати 0 та 1 і 2. Таким чином, формуються два зв'язки, при цьому старша цифра, яка залишається поза розрядною сіткою виштовхується і не враховується. Схожий спосіб з використанням надлишкового кодування описано в публікації [9-13]. Отже, існує число А:

$$A = a_i * 3^i + a_{i-1} * 3^{i-1} + \dots + a_1 * 3 + a_0,$$

де а –цифра від 0 до 2, і – кількість розрядів в числі А.

Сформуємо зв'язки за допомогою зсуву вліво. Після зсуву числа, отримуємо число As:

$$As = a_{i-1} * 3^i + a_{i-2} * 3^{i-1} + \dots + a_0 * 3 + e,$$

де е – набуває значення 0, 1 або 2.

Залежно від значення е, отримуємо 3 різних числа, з якими буде утворені зв'язки. Аналогічно до формування зв'язків для першої топології, існують 3 числа В₀, В₁, В₂ котрі формують зв'язок з А, опишемо загальною формулою для всіх трьох випадків для В_i, в якому е набуває значення 0, 1, 2 (рис. 5, табл. 6):

$$B_i = e * 3^i + a_i * 3^{i-1} + \dots + a_2 * 3 + a_1.$$

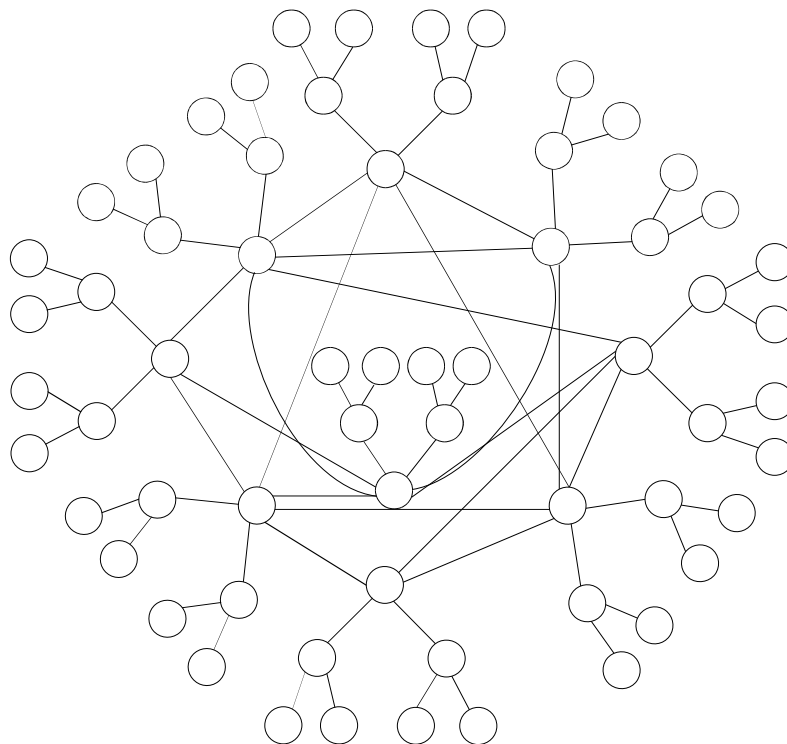


Рис. 5. Двійкові дерева, пов'язані між собою трійковими дебруйнівськими зв'язками

Таблиця 6

Характеристики першої запропонованої топології

К-сть вузлів	Діаметр	Ступінь	Середній діаметр	Ціна	Топологічний графік	SD
21	5	4	3,385711	420	1,69286	20
63	6	7	4,081413	2646	1,16612	42
189	7	8	4,849544	10584	1,21239	56
567	8	8	5,654383	36288	1,4136	64
1701	9	8	6,515968	122472	1,62899	72
5103	10	8	7,423336	408240	1,85583	80

Четверта запропонована топологія: **Декартовий добуток двійкового дерева та графу де Бруйна з двійковими зв'язками.**

Особливістю декартового добутку, як способу синтезу нової топологічної організації є те що, якщо відомо, яким чином зростають показники, або яка залежність існує між рангом (етапом масштабування) топології та показниками. Таким чином, можна аналітично передбачити які характеристики матиме отримана топологія на будь-якому з етапів масштабування. З цього випливає інша особливість, якщо топологія А та топологія В мають сталу ступінь, то утворена топологія С також матиме сталу ступінь на будь-якому з кроків. У результаті, утворена топологія на основі А та В може мати покращені характеристики, зберігаючи відмовостійкі властивості, які нас цікавлять. А саме, збереження достатньо невеликого ступеня в інтервалі від 4 до 6, та зменшення діаметру за рахунок створення додаткових зв'язків (рис. 6, табл. 7).

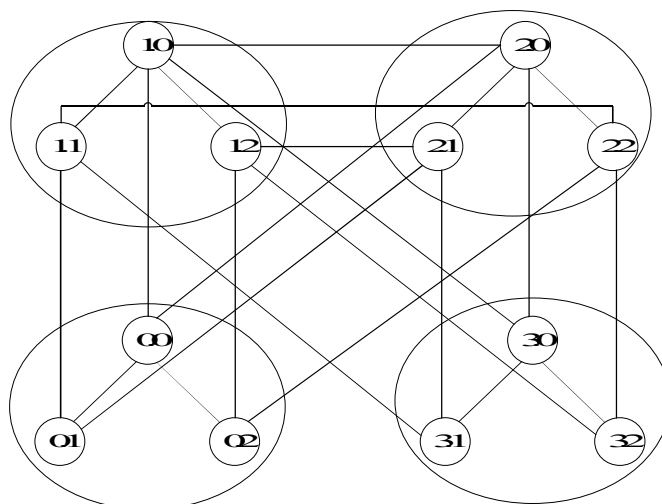


Рис. 6. Двійкові дерева помножені на граф де Бруйна з двійковими зв'язками

Таблиця 7

Характеристики декартового добутку двійкового дерева та графу де Бруйна з двійковими зв'язками

К-сть вузлів	Діаметр	Ступінь	Середній діаметр	Ціна	Топологічний графік	SD
12	4	5	2,19697	240	0,87879	20
56	7	7	3,703896	2744	1,05827	49
240	10	7	5,513354	16700	1,57524	70
992	13	7	7,621867	90272	2,17768	91
4032	16	7	9,986001	451584	2,85314	112

Зауважимо, що в даному способі ступінь дерева на кожному кроці масштабування менше на один, ніж ступінь графа де Бруйна.

П'ята запропонована топологія: **Трійкове дерево з трійковими дебруйнівськими зв'язками на кожному ярусі дерева.**

Підхід аналогічний до використаного у другій топології, але замість двійкових дебруйнівських зв'язків, застосовані трійкові дебруйнівські зв'язки. Оскільки дерево трійкове, то на кожному ярусі буде $3^r - 1$ вершин. Також необхідно використовувати саме трійкові дебруйнівські зв'язки для забезпечення зв'язків вершин на ярусі. За схожим до другої топології принципом, вузол p , знаходиться на ярусі $\lceil \log_3(p) \rceil = m$. Потім для $3m$ вершин встановлюються трійкові дебруйнівські зв'язки (рис. 7, табл. 8).

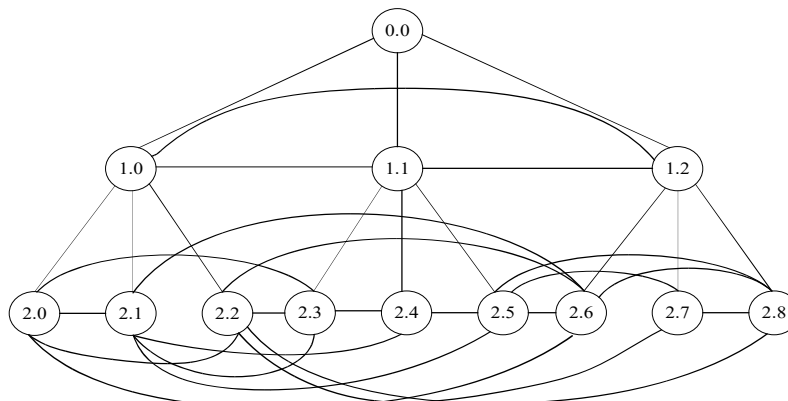


Рис. 7. Трійкове дерево з трійковими дебруйнівськими зв'язками на кожному ярусі дерева

Таблиця 8

Характеристики трійкового дерева з трійковими дебруйнівськими зв'язками

К-сть вузлів	Діаметр	Ступінь	Середній діаметр	Ціна	Топологічний трафік	SD
8	3	4	2	96	1	12
26	5	6	3,42462	780	1,14154	30
80	7	6	5,12405	3360	1,70802	42
242	9	6	6,98141	13068	2,32714	54
728	11	6	8,91851	48048	2,97284	66
2186	13	6	10,89222	170508	3,63074	78
6560	15	6	12,88165	590400	4,29388	90

Результати. На рис. 8-13 зображені діаграми топологічних характеристик залежно від кількості вершин. Топології, які розглянуті вище, показані на графіках у відповідному форматі:

Кластери з двійкових дерев, що ієрархічно об'єднані за принципом де Бруйна

1. Binary Tree Clusters with Hierarchical Binary DeBrujin

Двійкове дерево з горизонтальними дебруйнівськими зв'язками.

2. Binary Tree with horizontal DeBrujin connections

Двійкові дерева, пов'язані між собою трійковими дебруйнівськими зв'язками.

3. Binary Tree Clusters with Hierarchical Ternary DeBrujin

Декартовий добуток двійкового дерева та графу де Бруйна з двійковими зв'язками.

4. Cartesian Product of Binary Tree and Binary DeBrujin

Трійкове дерево з трійковими дебруйнівськими зв'язками на кожному ярусі дерева.

5. Ternary Tree with horizontal ternary DeBrujin connections

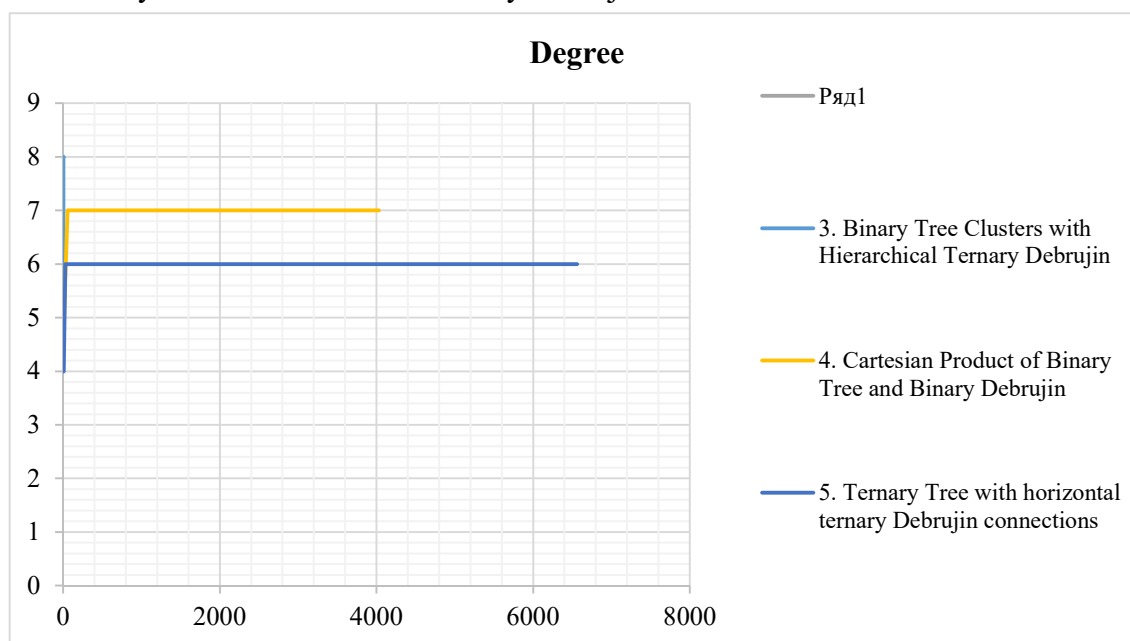


Рис. 8. Графік росту ступеня системи залежно від кількості вузлів

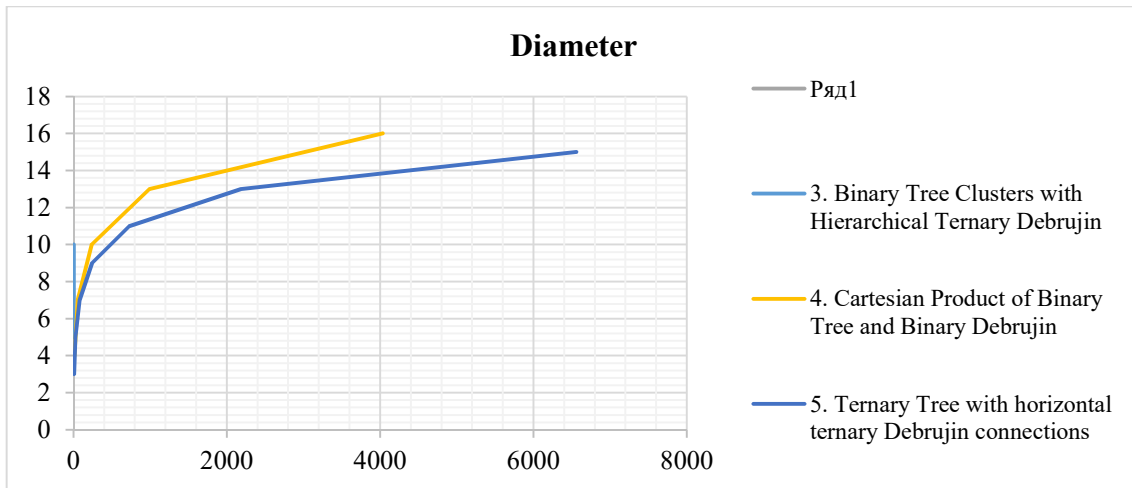


Рис. 9. Графік росту діаметра системи залежно від кількості вузлів

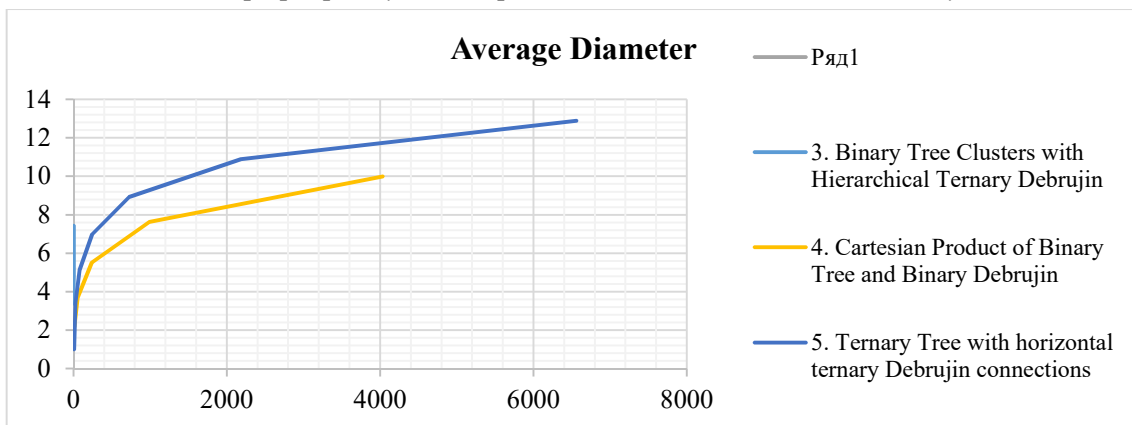


Рис. 10. Графік росту середнього діаметра системи залежно від кількості вузлів

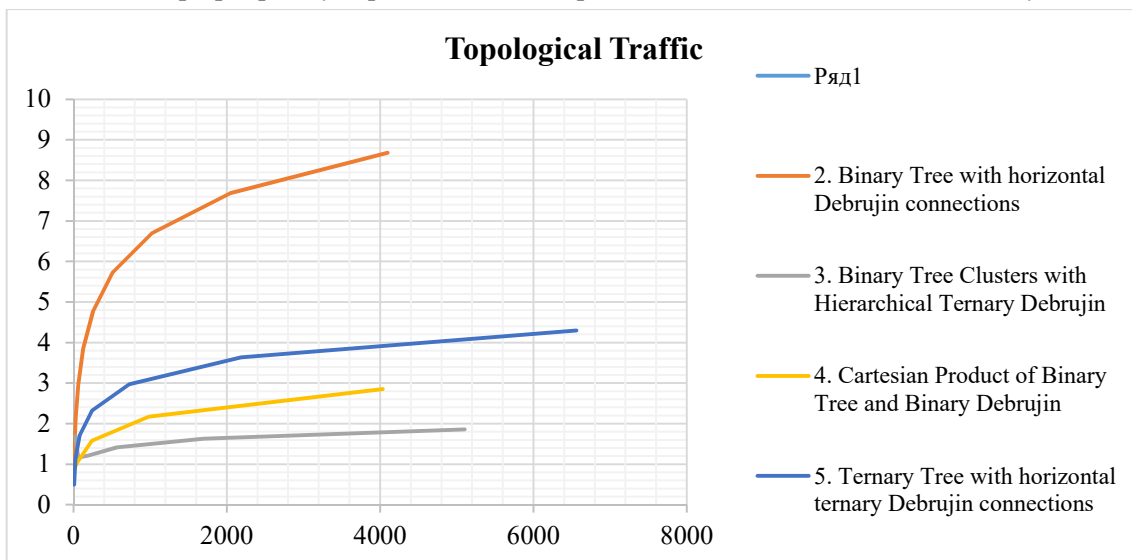


Рис. 11. Графік росту топологічного трафіку системи залежно від кількості вузлів

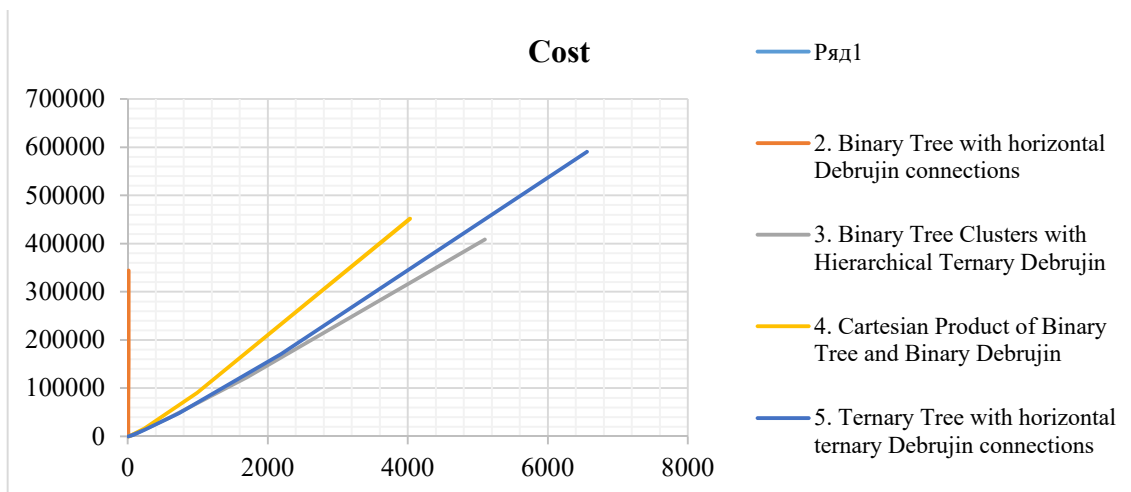


Рис. 12. Графік росту ціни системи залежно від кількості вузлів

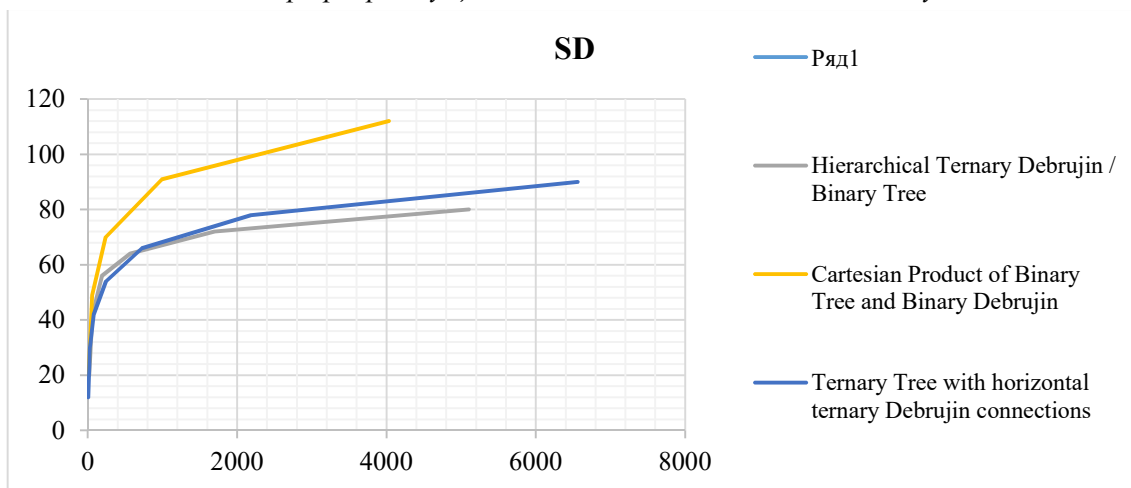


Рис. 13. Графік росту параметру SD системи залежно від кількості вузлів

З графіків можна побачити, як топологічні характеристики представлених організацій змінюються з ростом кількості вершин. Покажемо аналітичні формули для приросту параметрів залежно від кроку масштабування (табл. 9).

Таблиця 9

Таблиця росту топологічних характеристик залежно від кроку масштабування

	К-сть вершин	Ступінь	Діаметр	SD
1	$7 \cdot 2^R$	6	$4+R$	$6(4+R)$
2	2^{R-1}	4	$1+2R$	$4(1+2R)$
3	$7 \cdot 3^R$	8	$4+R$	$8(4+R)$
4	$2^{R \cdot (2^R - 1)}$	7	$1+3R$	$7(1+3R)$
5	3^{R-1}	6	$1+2R$	$6(1+2R)$

Висновки відповідно до статті. У роботі було розглянуто п'ять топологічних організацій та їх характеристики. В ході розробки способу синтезу нових топологій було встановлено залежність зростання діаметру від обраного способу. Показано, що для ієрархічного підходу, в якому в якості кластеру використовується дерево, а зв'язки встановлюються за допомогою дебруйнівських зсувів, діаметр зростає вдвічі повільніше в порівнянні з деревом із додатковими зв'язками. Також усі топології нарощують кількість вузлів, як ступінь основи системи кодування послідовності де Бруйна.

Перевагами даного підходу є, по-перше, лінійний приріст SD-параметра. Це свідчить про те, що один із показників залишається сталим протягом усіх кроків масштабування, в цьому випадку це показник ступеню. Стала ступінь дозволяє заздалегідь передбачувати

яка максимальна кількість зв'язків буде на будь-якому кроці, і передбачити витрати на мережеве обладнання. Також, суттєво зменшився показник діаметру для розглянутих топологій, у порівнянні з деревом.

Існують деякі недоліки, а саме показник топологічного трафіку має значення більше ніж 1, що стверджує про можливе утворення «заторів» пакетів на вузлах. Можливим варіантом покращення показників є використання модифікованих дерев, тобто поєднання одночасно трьох і більше топологій.

Список використаних джерел

1. Alverson R., Duncan Roweth, Larry Kaplan. The gemini system interconnect. *2010 18th IEEE Symposium on High Performance Interconnects*. 2010. Pp. 83-87.
2. Kim John, Wiliam J. Dally, Steve Scott, and Dennis Abts. Technology-driven, highly-scalable dragonfly topology. *2008 International Symposium on Computer Architecture*. 2008. Pp. 77-88.
3. Ajima, Yuichiro, Shinji Sumimoto, and Toshiyuki Shimizu. Tofu: A 6D mesh/torus interconnect for exascale computers. *Computer*. 2009. № 11. Pp. 36-40.
4. Guan K. C., Chan V. W. S. Cost-efficient fiber connection topology design for metropolitan area WDM networks. *OSA Journal of Optical Communications and Networking*. 2009. Т. 1, № 1. С. 158-175.
5. Ganesan Elango, Dhiraj K. Pradhan. The hyper-debruijn networks: Scalable versatile architecture. *IEEE Transactions on parallel and distributed systems*. 1993. № 4.9. Pp. 962-978.
6. Dürr F. (2016). A Flat and Scalable Data Center Network Topology Based on De Bruijn Graphs. arXiv preprint arXiv:1610.03245.
7. Kamal, Md Sarwar, et al. De-Bruijn graph with MapReduce framework towards metagenomic data classification. *International Journal of Information Technology*. 2017. № 9.1. Pp. 59-75.
8. Peng G., Ji P., Zhao F. A novel codon-based de Bruijn graph algorithm for gene construction from unassembled transcriptomes. *Genome Biol*. 2016. № 17. Pp. 232. URL: <https://doi.org/10.1186/s13059-016-1094-x>.
9. Olexandr G., Rehida P., Volokyta A., Loutskii H., Think V.D. Routing Method Based on the Excess Code for Fault Tolerant Clusters with InfiniBand. *Advances in Computer Science for Engineering and Education II. ICCSEEA 2019. Advances in Intelligent Systems and Computing*. 2020. Vol. 938. Springer, Cham.
10. Honcharenko O., Volokyta A., Loutskii H. Fault-tolerant topologies synthesis based on excess code usign the latin square. *The International Conference on Security, Fault Tolerance, Intelligence ICSFTI2019, Ukraine, Kyiv, 14-15 May, 2019. Kyiv, 2019*. Pp. 72-81.
11. Loutskii H., Volokyta A., Rehida P., Goncharenko O. Using excess code to design fault-tolerant topologies. *Technical sciences and technologies*. 2019. № 1(15). Pp. 134-144. DOI: [https://dx.doi.org/DOI: 10.25140/2411-5363-2019-1\(15\)-134-144](https://dx.doi.org/DOI: 10.25140/2411-5363-2019-1(15)-134-144).
12. Loutskii H., Volokyta A., Rehida P., Honcharenko O., Think V.D. (2021) Method for Synthesis Scalable Fault-Tolerant Multi-level Topological Organizations Based on Excess Code. *Advances in Computer Science for Engineering and Education III. ICCSEEA 2020. Advances in Intelligent Systems and Computing / Hu Z., Petoukhov S., Dychka I., He M. (eds.). Vol. 1247. Springer, Cham*. https://doi.org/10.1007/978-3-030-55506-1_32.
13. Loutskii H., Volokyta A., Rehida P., Honcharenko O., Ivanishchev B., Kaplunov A. Increasing the fault tolerance of distributed systems for the Hyper de Bruijn topology with excess code. *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*. Kyiv, 2019. Pp.1-6. DOI: 10.1109/ATIT49449.2019.9030487.

References

1. Alverson, Robert, Duncan Roweth, and Larry Kaplan. (2010). The gemini system interconnect. In *2010 18th IEEE Symposium on High Performance Interconnects* (pp. 83-87). IEEE.
2. Kim, J., Wiliam J. Dally, Steve Scott, and Dennis Abts. (2008). Technology-driven, highly-scalable dragonfly topology. In *2008 International Symposium on Computer Architecture* (pp. 77-88). IEEE.
3. Ajima, Yuichiro, Shinji Sumimoto, and Toshiyuki Shimizu. (2009). Tofu: A 6D mesh/torus interconnect for exascale computers. *Computer*, 11, pp. 36-40.

4. Guan, K. C., Chan, V. W. S. (2009). Cost-efficient fiber connection topology design for metropolitan area WDM networks. *OSA Journal of Optical Communications and Networking*, 1(1), pp. 158-175.
5. Ganesan, Elango, and Dhiraj K. Pradhan. (1993). The hyper-debruijn networks: Scalable versatile architecture. *IEEE Transactions on parallel and distributed systems*, 4.9, pp. 962-978.
6. Dürr, F. (2016). A Flat and Scalable Data Center Network Topology Based on De Bruijn Graphs. arXiv preprint arXiv:1610.03245.
7. Kamal, Md Sarwar, et al. (2017). De-Bruijn graph with MapReduce framework towards metagenomic data classification. *International Journal of Information Technology*, 9.1, pp. 59-75.
8. Peng, G., Ji, P. & Zhao, F. (2016). A novel codon-based de Bruijn graph algorithm for gene construction from unassembled transcriptomes. *Genome Biol.* 17, p. 232. <https://doi.org/10.1186/s13059-016-1094-x>.
9. Olexandr, G., Rehida, P., Volokyta, A., Loutskii, H., Thinh, V. D. (2020). Routing Method Based on the Excess Code for Fault Tolerant Clusters with InfiniBand. *Advances in Computer Science for Engineering and Education II. ICCSEE 2019. Advances in Intelligent Systems and Computing*, 938. Springer, Cham
10. ncharenko, O., Volokyta, A., Loutskii, H. (2019). Fault-tolerant topologies synthesis based on excess code usign the latin square. *The International Conference on Security, Fault Tolerance, Intelligence ICSFTI2019* (pp. 72-81).
11. Loutskii, H., Volokyta, A., Rehida, P., Goncharenko, O. (2019). Using excess code to design fault-tolerant topologies. *Technical sciences and technologies*, 1(15), pp. 134–144. DOI: [https://dx.doi.org/DOI: 10.25140/2411-5363-2019-1\(15\)-134-144](https://dx.doi.org/DOI: 10.25140/2411-5363-2019-1(15)-134-144).
12. Loutskii, H., Volokyta, A., Rehida, P., Honcharenko, O., Thinh, V. D. (2021). Method for Synthesis Scalable Fault-Tolerant Multi-level Topological Organizations Based on Excess Code. In Hu Z., Petoukhov S., Dychka I., He M. (eds.), *Advances in Computer Science for Engineering and Education III. ICCSEE 2020. Advances in Intelligent Systems and Computing*, 1247. Springer, Cham. https://doi.org/10.1007/978-3-030-55506-1_32.
13. H. Loutskii, A. Volokyta, P. Rehida, O. Honcharenko, B. Ivanishchev and A. Kaplunov (2019). Increasing the fault tolerance of distributed systems for the Hyper de Bruijn topology with excess code. *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*. (pp. 1-6). DOI: 10.1109/ATIT49449.2019.9030487.

UDC 004.75

Volodymyr Rusinov, Bohdan Ivanishchev, Andrii Antoniuk, Artem Volokyta, Heorhii Loutskii

METHODS OF TOPOLOGICAL ORGANIZATION SYNTHESIS BASED ON DE BRUIJN CODE TRANSFORMATIONS

Urgency of the research. Today, topological organizations such as trees, fat trees, 3D-mesh, 6D-tor, DragonFly are used as a way of network organization in data centers. All these topologies have fault-tolerant properties due to the presence of several alternative routing paths. Therefore, the task of developing new topologies that have many balanced routes (one cost path) is relevant. One of the best options for research on tree topology improvement is the combination with de Bruijn graphs in different number systems, which allows varying the degree of vertices, diameter and number of alternative paths for routing. This paper proposes a new way to obtain specified characteristics when combining tree-based topologies with a de Bruijn graph.

Target setting. When creating high-performance computing systems, it is important to develop their topological organizations in advance, which allows in time to anticipate the characteristics of the system.

Actual scientific researches and issues analysis. At present there are a number of works devoted to topologies based on de Bruijn sequences. There is a detailed review of the synthesis of hybrid topologies based on de Bruijn graphs and hypercube.

Uninvestigated parts of general matters defining. The possibility of using hybrid topologies based on de Bruijn sequences and different types of trees remains unexamined at present.

The research objective. The tasks are to develop a way to synthesize new topologies based on trees and de Bruijn sequences and to analyze the characteristics of the synthesized topological organizations for different levels of fault tolerance.

The statement of basic materials. The synthesis of five topologies based on trees and de Bruijn sequences is described. A comparative analysis of all synthesized topologies is performed.

Coclusions. An analysis of the characteristics is performed, the main advantages and disadvantages of the proposed topological structures are highlighted, and suggestions for their improvement are made.

Keywords: de Bruijn graph, topology, fault tolerance.

Fig.: 13. Table: 9. References: 13.

Луцький Георгій Михайлович – доктор технічних наук, професор, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (просп. Перемоги, 37, м. Київ, 03056, Україна).

Loutskii Heorhii – Doctor of Technical Sciences, Professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (37 Pobedy Av., 03056 Kyiv, Ukraine).

E-mail: georgijluckyj80@gmail.com

ORCID: <http://orcid.org/0000-0002-3155-8301>

Scopus Author ID: 16473143100

Волокита Артем Миколайович – кандидат технічних наук, доцент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (просп. Перемоги, 37, м. Київ, 03056, Україна).

Volokyta Artem – PhD, Associate Professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (37 Pobedy Av., 03056 Kyiv, Ukraine).

E-mail: artem.volokita@kpi.ua

ORCID: <http://orcid.org/0000-0001-9069-5544>

Scopus Author ID: 54421406500

Антонюк Андрій Іванович – кандидат технічних наук, доцент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (просп. Перемоги, 37, м. Київ, 03056, Україна).

Antonuk Andrii – Ph.D., Associate professor, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (37 Pobedy Av., 03056 Kyiv, Ukraine).

E-mail: ant5298g@gmail.com

ORCID: <http://orcid.org/0000-0002-7134-6100>

Іванішев Богдан Вячеславович – асистент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (просп. Перемоги, 37, м. Київ, 03056, Україна).

Ivanishchev Bohdan – assistant, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (37 Pobedy Av., 03056 Kyiv, Ukraine).

E-mail: ivanishchev@comsys.kpi.ua

ORCID: <http://orcid.org/0000-0003-4726-0689>

Русінов Володимир Володимирович – студент, кафедра обчислювальної техніки, Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» (просп. Перемоги, 37, м. Київ, 03056, Україна).

Rusinov Volodymyr – student, Department of Computer Engineering, National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (37 Pobedy Av., 03056 Kyiv, Ukraine).

E-mail: volodymyr.r.v@ukr.net

ORCID: <https://orcid.org/0000-0002-4362-0248>