УДК 004.72

DOI: 10.18372/2073-4751.80.19784

Гончаренко О.О.,
orcid.org/0000-0002-9086-6988,
e-mail: alexandr.ik97@ukr.net,
Волокита А.М.,к.т.н.,
orcid.org/0000-0001-9069-5544,
e-mail: artem.volokita@kpi.ua

МЕТОД СИНТЕЗУ ВІДМОВОСТІЙКИХ ТОПОЛОГІЙ З ІМПЛІЦИТНИМИ КЛАСТЕРАМИ НА ОСНОВІ ПЕРЕТВОРЕНЬ ДЕ БРУЙНА В НАДЛИШКОВИХ СИСТЕМАХ ЧИСЛЕННЯ

**Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»**

Вступ

Важливим аспектом, що характеризує сучасні найпродуктивніші комп'ютерні системи (КС), є використання надвеликої кількості ядер – від сотні тисяч до десяти мільйонів. Таке велике число обчислювальних елементів, які об'єднані в мережу з певною топологією, дає змогу отримати широкі можливості для паралельної обробки, але побічним ефектом цього є наявність в системі великої кількості потенційних точок відмов. Відомі методи забезпечення відмовостійкості, такі як резервування, дозволяють вирішити цю проблему, втім їх використання в загальному вигляді веде до суттєвого удорожчання системи, що для суперкомп'ютера є критичним.

Не менш важливим питанням є і питання ефективності, оскільки не дивлячись на високу номінальну продуктивність, сучасні супер-комп'ютери демонструють відносно посередню ефективність. Згідно до даних рейтингу *TOP* 500 на листопад 2023 року [1] сучасні суперкомп'ютери зазвичай побудовані на основі *dragonfly*-подібних топологій та жирних дерев [2-18]. Окремо варто зазначити, що ці топології часто мають великий рівень надлишковості для забезпечення відмовостійкості.

Таким чином, актуальною є задача синтезу спеціалізованих топологій [19, 20], що дозволить закласти в систему надлишковість ще на етапі конструювання, керуючи при цьому топологічними параметрами і досягаючи компромісу між відстанню до найвіддаленіших вузлів та щільністю комутацій. Втім варто зазначити, що ідея керування характеристиками мережі через топологію не є новою. Існує багато наукових праць, присвячених синтезу топологій для задач обчислювальної техніки і, зокрема, високопродуктивних обчислень [2-18].

Окремою проблемою є те, що, не дивлячись на актуальність питання відмовостійкості, саме їй в існуючих розробках присвячено небагато уваги. Це породжує потребу в нових методах та актуалізує пошук таких рішень, які б дозволили досягти компромісу між основними критеріями ефективності КС: ефективністю передачі даних, вартістю та стійкістю до відмов.

Постановка задачі

Завдання полягає в розробці методу синтезу відмовостійких топологій на основі перетворень де Бруйна [2] та надлишкового коду [19], розробці способу формування кластерів у таких графах, а також у розробці способу моделювання живучості [20] таких топологій, який би дозволив врахувати навантаження на окремі вузли системи.

Для моделювання топологій розроблено додаток на мові *Python* з використанням бібліотеки *NetworkX*, що дозволяє синтезувати топології, вимірювати їх топологічні характеристики та досліджувати живучість статистичними методами, представляючи результат у вигляді імовірності p, що при заданому числі відмов елементів (вузлів графа) система зберігатиме працездатність (граф її мережі залишатиметься зв'язним).

В якості основних критеріїв ефективності пропонується розглядати: 1) мультиплікативна характеристика ступеня та діаметра (*SD*) як комплексна оцінка ефективності передачі даних та вартості, 2) живучість як основний показник відмовостійкості.Мета

Метою даної роботи є підвищення ефективності комп'ютерних систем за рахунок синтезу нових топологій, що дозволить досягти компромісу між ефективністю передачі даних (показник діаметру), вартістю (показник ступеня) та відмовостійкістю (показник живучості).

Основна частина

Сучасні суперкомп'ютери використовують ряд топологій (табл. 1), проте найпопулярнішими з них є жирне дерево та dragonfly. Існує велика кількість їх варіацій. Ключовими їх особливостями можна назвати малий діаметр та високу надлишковість, яка дозволяє забезпечити вкрай високу відмовостійкість. Втім, ці два типи топологій мають один суттєвий недолік: їх надлишковість є занадто високою, що робить вартість надзвичайно високою.

Таблиця 1. Порівняння топологій

Топологія (порядку r)	Закон масштабування $N(r)$	Ступінь $S(r)$	Діаметр $D(r)$
Гіперкуб	Експоненційний	Пропорційна r	Пропорційний r
Тор та тороїдальні графи (n-вимірні)	Степеневий (за степенем n)	Пропорційна n	Пропорційний r
Leaf/spine	r	$r/2$	3
Fat Tree (неблокуюче, 4-рівневе)	$5r^2/_4$	r	5
Dragonfly (внутрішні та зовнішні зв'язки розподілені 50/50)	$r^3 + 2r^2 + 2r + 1$	r	3
Dragonfly+ (внутрішні та зовнішні зв'язки розподілені 50/50)	$r^3/_2 + r$	r	5

Одним із перспективних рішень є використання графів зсувів-вставок, що будуються на послідовностях де Бруйна [2] в певних системах числення (СЧ). Втім, класичний бінарний чи n-арний код, хоч і дозволяє досягти переваги за параметром SD, має свої недоліки: число альтернативних маршрутів в такому графі обмежено, що потребує удосконалення підходу.

Рішенням може бути надлишковий код, число цифр k алфавіту якого є більшим за основу числення b. Таким чином, кілька кодів можуть мати однакове числове значення. В контексті топології це означає, що деякі вершини графа «склеюються», формуючи одну логічну вершину. Рис. 1 наочно ілюструє цю ідею для графа де Бруйна на основі коду $T01_2$.

В попередніх роботах [19] було досліджено поняття шаблону – 2-розрядного коду, який дає змогу «переходити» між альтернативними представленнями числа. Проте, знаючи число шаблонів та їх «потужність», можна також передбачити загальне число альтернативних представлень, – а також запропонувати спосіб формування логічних вершин.

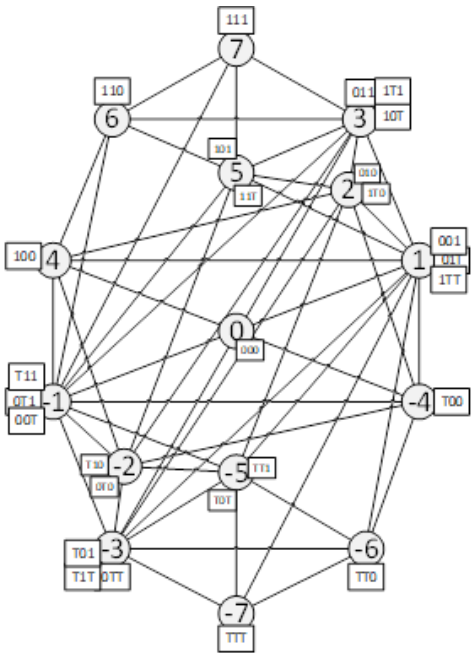

Рис. 1. Формування логічних вершин для надлишкового графа де Бруйна

Таблиця 2. Аналвз 2-х розрядних кодів

Знач.	$01T_2$	0123_2	$T012_2$	$T012_3$	01234_2
12					44
11					43
10					34, 42
9		33			33, 41
8		32		22	24, 32, 40
7		31, 23		21	23, 31
6		30, 22	22	20	14, 22, 30
5		21, 13	21	12, 2T	13, 21
4		20, 12	20, 12	11	04, 12, 20
3	11	11, 03	11, 2T	10	03, 11
2	10	10, 02	10, 02	02, 1T	02, 10
1	01, 1T	01	01, 1T	01	01
0	00	00	00, T2	00	00
-1	0T, T1		0T, T1	T2, 0T	
-2	T0		T0	T1	
-3	TT		TT	T0	
-4				TT	

Аналізуючи 2-розрядні коди (табл. 2), можна помітити важливу річ: коди, які починаються з однієї цифри, слідують одне за одним, не перекриваючись. Таким чином, виникає ланцюг, що не містить в собі надлишковості. В той же час ці ланцюги перекриваються одне з одним, накладаючись із кроком b (починаючи від коду з найменшим значенням). Це дозволяє описати надлишкові властивості 2-розрядної надлишкової СЧ через ланцюгові схеми. Приклад такої схеми на рис. 2.

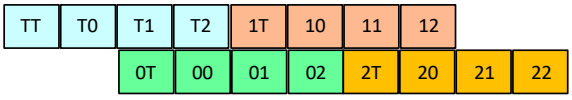

Рис. 2. Ланцюгова схема коду $T012_2$

Властивості ланцюгів, очевидно, витікають із теорії позиційних систем числення. Таким чином, нескладно розширити дану схему на випадок коду довільної розрядності. Для цього позначимо число альтернативних кодових представлень деякого довільного числа $v \in \mathbb{Z}$ як α(v). Нехай відомими є значення $\alpha_r(v)\ \forall\ v \in \mathbb{Z}$. Тоді для $\alpha_{r+1}(v)$ справедливою є наступна формула:

$$\alpha_{r+1}(v) = \sum_{i\in\mathbb{Z}} \alpha_r(v - ib^r). \qquad (1)$$

Доведемо цю формулу. Нехай є деякий r-розрядний код $\mu = a_{r-1}a_{r-2}\dots a_0$, такий, що при доповненні його старшою цифрою у представлятиме число v (позначимо це як $V(\mu) = v$. Очевидно, що шаблонний перехід по будь-якому із шаблонів в μ дає рівно α_r представлень. Таким чином, єдиний шаблон, перехід по якому не враховано в $\alpha_r(V(\mu))$ – це шаблон ya_{r-1}.

Втім, проблемою є те, що кількість альтернативних представлень μ не може бути визначена виключно через один шаблон. Так, якщо $a_{r-1}a_{r-2}$ є шаблоном з альтернативними представленнями, то, окрім ya_{r-1}, в деяких альтернативних формах μ з'являтимуться інші, альтернативні шаблони.

Втім, цей фактор можна врахувати, якщо перейти від кодів до їх значень. Згідно з властивостями позиційної СЧ, значення yμ можна представити наступною формулою:

$$V(y\mu) = yb^r + V(\mu). \qquad (2)$$

Аналогічно, будь-який шаблонний перехід в системі числення з алфавітом A може бути описаний так: $\forall\, T = yx, y \in A, x \in A\ \exists\, T'_{i\in\mathbb{Z}} = y'x' \leftrightarrow \exists\, y' = (y + i)\ \in A, \exists\, x' = (y - ib) \in A$.

Очевидно, що незалежно від внутрішніх переходів в μ значення $V(\mu) = const$. Таким чином, формула (2) є справедливою для всіх можливих ya_{r-1}. Одночасно із тим, при переході по ya_{r-1} сам у змінюється у відповідності до можливих цифр алфавіту (з кроком $i \in \mathbb{Z}$), а $V(\mu') = V(\mu) - i * b * b^{r-1} = V(\mu) - ib^r$. Відповідно, для кожної альтернативи у можна отримати число альтернативних представлень для μ', що є рівним $\alpha_r\big(V(\mu')\big) = \alpha_r(V(\mu) - ib^r) = \alpha_r(v - ib^r)$.

Дана формула дозволяє аналітично передбачувати кількість альтернативних представлень, тим самим прогнозуючи кількість вершин з однаковим номером. Це дозволяє виявити всі представлення одного числа і впорядкувати їх у багатовимірну матрицю таким чином, щоб кожному шаблонному переходу відповідав перехід по одному з вимірів матриці (але не навпаки). Таким чином, вершини з однаковим номером можуть бути сформовані в гіперкуб з надлишковою основою $t = \left\lceil {}^{k}/_{b} \right\rceil$.

Таким чином, логічні вершини в топології є імпліцитними (неявними) кластерами, невидимими ззовні (з точки зору кодових перетворень), проте доступними для таких операцій як резервування чи обхід відмов при маршрутизації. Також це потенційно дозволяє перерозподілити трафік, що передаватиметься по мережі, і тим самим уникати перевантаження окремих вузлів при виникненні відмови.

Запропонований *спосіб формування кластерів* містить наступні кроки:

1. Розрахувати всі альтернативні представлення числа.

2. Розташувати їх у багатовимірній матриці так, щоб індекс відповідного виміру визначався як $\sigma_i = a_{i+1} mod\ t$. При цьому в матриці можуть з'являтись порожні елементи, для яких немає відповідного коду.

3. Досягти того, щоб переходу по виміру відповідав шаблонний перехід за допомогою перестановки елементів матриці за правилом $\sigma_i' = \sigma_i + \delta\sigma_{i+1}$, $\delta = b\ mod\ t$.

4. Зв'язати відповідні альтернативні представлення (в гіперкуб із надлишковою основою), пропускаючи порожні елементи.

Використовуючи формулу (1), можна виконати аналіз характеристик різноманітних кодів. Окрему увагу тут слід звернути на коди з потужністю алфавіту 4 та основою 2, такі як $T012_2$, оскільки структура їх базового розподілу α_2 не містить «прогалин» в центрі (табл. 2). Таким чином, при накладанні розподілів за формулою (1), центральна частина результуючого розподілу матиме потужність $\alpha_{r,max} = t^r$, що є ідеальним для гіперкуба.

Маючи даний спосіб в якості основи, можливо модифікувати класичний метод **синтезу топологій з використанням кодових перетворень**.

Етапи методу:

1. Обрання позиційної системи числення (СЧ), що складається із множини цифр (алфавіту) A потужності k та основи числення *b*, таких, що *k>b*.

2. Обрання рангу топології $r = log_k(N_{desired} - 1) + 1$, де $N_{desired}$ – бажане число вузлів системи.

3. Формування множини кодів *C*, що має потужність $N = k^r$ і містить всі можливі r-розрядні коди $a_{r-1}a_{r-2} \dots a_1a_0$ обраної системи числення.

4. Формування множини вершин графа *G*, елементи якої мають взаємно-однозначну відповідність із елементами множини *C*.

5. Обрання бажаних кодових перетворень $f_1, f_2, \dots, f_m$ в заданій системі числення.

6. Формування множини E ребер графа G (C, E) за правилом $\forall\, i = 1 \dots m\ \&\ \forall\, c \in C\colon\ f_i(c) = c'_i,\ c'_i \in C \rightarrow\ \exists\, e_{c \leftrightarrow c'_i} \in E$

7. Формування множини додаткових ребер E' для формування імпліцитних кластерів

В табл. 3 продемонстровано перетворення на основі послідовностей де Бруйна, модифіковані для 2-розрядного коду $T012_2$.

Таблиця 3. Перетворення на основі послідовностей де Бруйна, мо-дифіковані для 2-розрядного коду T0122.

Код	Зсув вліво				Зсув вправо			
	T	0	1	2	T	0	1	2
TT	TT	0T	1T	2T	TT	T0	T1	T2
T0	TT	0T	1T	2T	0T	00	01	02
T1	TT	0T	1T	2T	1T	10	11	12
T2	TT	0T	1T	2T	2T	20	21	22
0T	T0	00	10	20	TT	T0	T1	T2
00	T0	00	10	20	0T	00	01	02
01	T0	00	10	20	1T	10	11	12
02	T0	00	10	20	2T	20	21	22
1T	T1	01	11	21	TT	T0	T1	T2
10	T1	01	11	21	0T	00	01	02
11	T1	01	11	21	1T	10	11	12
12	T1	01	11	21	2T	20	21	22
2T	T2	02	12	22	TT	T0	T1	T2
20	T2	02	12	22	0T	00	01	02
21	T2	02	12	22	1T	10	11	12
22	T2	02	12	22	2T	20	21	22

На основі цих перетворень може бути сформована нова топологія, яка зображена на рис. 3. Чорним кольором позначено вершини, що не мають інших представлень. Іншими кольорами – вершини,

що входять в імпліцитні кластери. Додаткові зв'язки (після формування кластерів) позначено штрихованими кольоровими лініями.

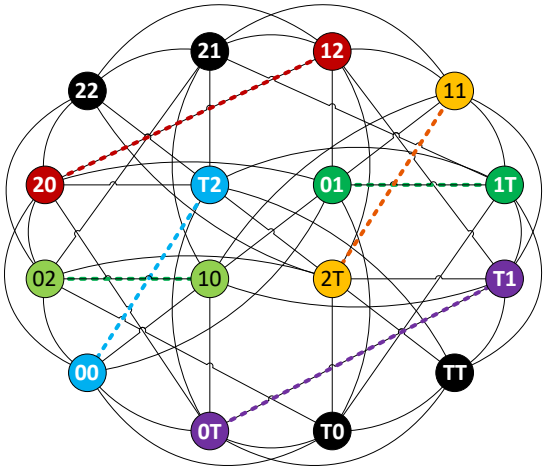

Рис. 3. Граф де Бруйна $T012_2$

Його топологічні характеристики представлено в табл. 4.

Таблиця 4. Топологічні характеристики графа де Бруйна $T012_2$

r	N	S	D	$\overline{D}$	T	E
2	16	8	2	1.533	0.438	0.733
3	64	10	3	2.170	0.479	0.520
4	256	11	4	2.867	0.552	0.384
5	1024	12	5	3.623	0.631	0.297
6	4096	13	6	4.432	0.709	0.238

Літерою T тут позначено топологічний трафік, літерою E – топологічну ефективність, розраховану за формулою:

$$E(G) = \frac{\sum_{i \neq j \in G} \frac{1}{d_{i,j}}}{N(N-1)}. \qquad (3)$$

Таким чином, запропонована топологія для 4096 вузлів має мультиплікативний критерії SD лише 78. Для порівняння, *dragonfly* з повнозв'язними кластерами на 16 вузлів та 15 зовнішніми зв'язками (3856 вузли) має ступінь 30 та діаметр 3 (SD = 90).

Наступним кроком є оцінка живучості топології. Класичний підхід передбачає присвоєння кожному ребру деякої імовірності відмови p та вилучення ребер із графа до втрати графом зв'язності [3]. Втім, недоліком такого підходу є залежність від заданих ззовні коефіцієнтів.

В даній роботі пропонується декілька модифікацій відомого способу, а саме:

1. Відмови зачіпають не ребра, а вузли. При відмові вузла всі його ребра вилучаються із графа.
2. Імовірність відмови p має визначатись із урахуванням характеристик графа.

Такий підхід дає змогу проаналізувати топологію з точки зору вразливості вузлів, які є більш складними та дорогими компонентами системи.

Для визначення імовірності p пропонується скористатись наступним припущенням: більш вразливими (відносно інших) є ті вузли, через які проходить більше трафіку. А більше трафіку проходить через ті вузли, які входять до більшого числа найкоротших маршрутів. Таким чином, для динамічного визначення імовірності відмови може бути використаний показник посередництва, що обчислюється за формулою:

$$c_b(v) = \sum_{s,t \in G} \frac{\sigma(s,t|v)}{\sigma(s,t)}, \qquad (4)$$

де, $\sigma(s, t)$ – число найкоротших шляхів між вершинами s та t, а $\sigma(s, t|v)$ – число найкоротших шляхів між вершинами s та t, що проходять через заданий вузол v.

Таким чином, імовірність відмови кожного вузла на кожному кроці моделювання можна обчислити наступним чином:

$$p_v(f) = \frac{c_b(v,f)}{\sum_{i \in G} c_b(i,f)}. \qquad (5)$$

Запропонований спосіб дослідження живучості топологій складається із наступних кроків:

Задається початковий граф G та число експериментів M.

Для всіх вершин топології розраховується імовірність відмови $p_n(f)$, де n – вершина графа, f – число вершин, що вже відмовили. Принцип розрахунку цієї імовірності називається потоком відмов.

На основі розрахованої імовірності обирається чергова вершина n, яка видаляється з графа разом зі своїми ребрами.

Якщо граф внаслідок відмови втратив зв'язність ($D = \infty$), даний експеримент завершується. Це означає, що топологія «вмирає», тобто, стає несправною.

Виконується вимірювання топологічних характеристик і повернення на крок 2.

Експеримент повторюється M разів, збираючи, таким чином, статистичні дані.

Характеристика живучості при цьому вимірюється для кожного f як відношення $M(f)$ експериментальних моделей, що «вижили» при цьому числі відмов, до загального числа експериментів M.

В табл. 4 представлено результати моделювання живучості. Отримані результати показують імовірність того, що при заданій кількості несправних вузлів граф зберігатиме зв'язність. Останній рядок таблиці показує максимальний відсоток відмов, при яких хоча би одна із експериментальних моделей зберігала зв'язність, а також – відповідне цьому значенню число вузлів.

Таблиця 4. Результати моделювання живучості

f, % від N	*Граф де Бруйна $T012_2$, N=256*	*Гіперкуб, N=256*	*Dragonfly, N=272*
10	1.00	1.00	1.00
20	1.00	1.00	1.00
30	1.00	0.99	0.93
40	0.97	0.92	0.34
50	0.89	0.68	-
60	0.49	0.15	-
70	0.06	-	-
80	0.01	-	-
f_{max}	80% (*N*=52)	67 % (*N*=85)	44 % (*N* = 153)

Висновки

В роботі виконано огляд сучасних топологій комп'ютерних систем, визначено їх характеристики. На підставі аналізу було визначено, що сучасні популярні рішення, такі як *dragonfly*, приділяють занадто високу увагу надлишковості та діаметру, при цьому їх вартість є зависокою, а можливості щодо використання надлишковості для відмовостійкості – обмеженими.

Проаналізовано надлишкові системи числення в контексті синтезу графів. Розглянуто формули, що дозволяють аналітично визначити число альтернативних представлень, і на їх основі запропоновано спосіб формування імпліцитних кластерів між вершинами з однаковими номерами. Таким чином, отримав розвиток метод синтезу топологій на основі перетворень де Бруйна в надлишкових системах числення.

Запропонований метод дозволив синтезувати топологію, що має кращий мультиплікативний критерій ступеня та діаметру (*SD*), ніж у топології *dragonfly*.

Також запропоновано новий спосіб дослідження живучості топологій на основі збору статистики та динамічного визначення імовірності відмови вузлів через коефіцієнт посередництва.

Проведено дослідження живучості топологій, встановлено, що запропонована топологія краще адаптується до відмов вузлів, тим самим забезпечуючи вищу імовірність виживання і цим підвищуючи відмовостійкість системи, що буде розроблена на її основі.

На основі запропонованих підходів розроблений додаток, що реалізує представлені способи, який може бути застосований розробниками комп'ютерних систем (в тому числі кластерів та суперкомп'ютерів) для дослідження топологій відповідних систем та пошуку рішень, що є задовільними з точки зору вимог до вартості, швидкості передачі даних та відмовостійкості.

Література

1. November 2023 | TOP500. URL: https://www.top500.org/lists/top500/2023/11.
2. Esfahanian, Hakimi. Fault-tolerant routing in debruijn comrnunication networks. IEEE Transactions on Computers, 1985. Vol. 100(9). P. 777–788.
3. Atchley S. et al. (2023, November). Frontier: Exploring Exascale The System Architecture of the First Exascale Supercomputer. *SC23: International Conference for High Performance Computing, Networking, Storage and Analysis* : proceedings, Denver, CO, USA, 11–17 November 2023 / SIGHPC, IEEE CS. New York, 2023. P. 1–16. DOI: 10.1145/3581784.3607089.
4. Aurora | Argonne Leadership Computing Facility. (n.d.). URL: https://www.alcf.anl.gov/aurora.
5. Eagle System Configuration. (n.d.). High-Performance Computing | NREL. URL: https://www.nrel.gov/hpc/eagle-system-configuration.html
6. Ajima Y. High-dimensional interconnect technology for the K computer and the supercomputer Fugaku. URL: https://www.fujitsu.com/global/about/resources/publications/technicalreview/topics/article005.html.
7. Documentation - Network and interconnect. (n.d.). URL: https://docs.lumi-supercomputer.eu/hardware/network/.
8. About | Leonardo pre-exascale supercomputer. (2024, February 21). Leonardo Pre-exascale Supercomputer. URL: https://leonardo-supercomputer.cineca.eu/about/#:~:text=Leonardo%20features%20a%20Dragonfly%2B%20topology,HPC%20application%20performance%20and%20scalability.
9. Stunkel C. B. et al. The high-speed networks of the Summit and Sierra supercomputers. *IBM Journal of Research and Development*. 2020. Vol. 64(3/4). P. 3–1.
10. MareNostrum 5. (n.d.). BSC-CNS. URL: https://www.bsc.es/ca/marenostrum/marenostrum-5
11. Morgan T. P. (2022, October 26). The NVSwitch fabric that is the hub of the DGX H100 SuperPOD. The Next Platform. URL: https://www.nextplatform.com/2022/03/23/nvidia-will-be-a-prime-contractor-for-big-ai-supercomputers/.
12. Wang T. et al. Rethinking the data center networking: Architecture, network protocols, and resource sharing. *IEEE access*. 2014. Vol. 2. P. 1481–1496.
13. Jain N. et al. Predicting the performance impact of different fat-tree configurations. *SC '17: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* : proceedings, Denver, CO, USA, 12–17 November 2017 / SIGHPC, IEEE CS. New York, 2017. P. 1–13. DOI: 10.1145/3126908.312696.
14. Ohring S. R. et al. On generalized fat trees. *9th international parallel processing symposium* : proceedings. Santa Barbara, CA, USA, 25–28 April 1995 / IEEE. 1995. P. 37–44. DOI: 10.1109/IPPS.1995.395911.
15. Zahavi, E. (2010). D-Mod-K routing providing non-blocking traffic for shift permutations on real life fat trees. CCIT Report, 776, 840.
16. Alizadeh M., Edsall T. On the data path performance of leaf-spine datacenter fabrics. *2013 IEEE 21st Annual Symposium on High-Performance Interconnects* : proceedings. San Jose, CA, USA, 21–23 August 2013 / IEEE. 1995. P. 71–74. DOI: 10.1109/IPPS.1995.395911.
17. Sabir E., Mamut A., Vumar E. The extra connectivity of the enhanced hypercubes. *Theoretical Computer Science*. 2019. Vol. 799. P. 22–31.
18. Shpiner A. et al. Dragonfly+: Low cost topology for scaling datacenters. In 2017 IEEE 3rd International Workshop on High-Performance Interconnection Networks in the Exascale and Big-Data Era (HiPINEB) (pp. 1-8). IEEE.
19. Loutskii H. et al. 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* : proceedings. Kyiv, Ukraine, 18–20 December 2019 / IEEE. 2019. P. 1–6. DOI: 10.1109/ATIT49449.2019.9030487.

20. Dodonov A., Lande D. Modeling the Survivability of Network Structures. URL: https://www.academia.edu/download/108489732/paper1.pdf.

Гончаренко О.О., Волокита А.М.

МЕТОД СИНТЕЗУ ВІДМОВОСТІЙКИХ ТОПОЛОГІЙ З ІМПЛІЦИТНИМИ КЛАСТЕРАМИ НА ОСНОВІ ПЕРЕТВОРЕНЬ ДЕ БРУЙНА В НАДЛИШКОВИХ СИСТЕМАХ ЧИСЛЕННЯ

Робота присвячена розробці методу синтезу топологій на основі перетворень де Бруйна в надлишкових системах числення, що дозволяє синтезувати відмовостійкі топології заданого порядку, в тому числі з імпліцитними кластерами. Також розроблено спосіб формування таких кластерів та спосіб дослідження живучості топологій з використанням динамічного визначення імовірності відмов на основі посередництва.

Запропонований комплексний підхід дозволяє синтезувати графи, що, з одного боку, містять в собі меншу надлишковість, а з іншого – мають вищу живучість за рахунок кращого використання наявної надлишковості, що дає змогу підвищити відмовостійкість з меншими витратами та забезпечити кращу ефективність.

Ключові слова: *топологія; ефективність; відмовостійкість; живучість; послідовності де Бруйна.*

Honcharenko O.O., Volokyta A.M.

METHOD FOR SYNTHESIS OF FAULT-TOLERANT TOPOLOGIES WITH IMPLICIT CLUSTERS BASED ON DE BRUIJN TRANSFORMATIONS IN REDUNDANT NUMERAL SYSTEMS

The work is devoted to the development of a method for synthesizing topologies based on de Bruijn transformations in redundant numeral systems, which allows synthesizing fault-tolerant topologies of a given order, including those with implicit clusters. A method for forming such clusters and a method for studying the survivability of topologies using dynamic determination of failure probability based on betweenness centrality are also developed.

The proposed comprehensive approach allows us to synthesize graphs that, on the one hand, contain less redundancy, and on the other hand, have higher survivability due to better use of the available redundancy, which allows us to increase fault tolerance with lower costs and ensure better efficiency.

Keywords*: topology; efficiency; fault tolerance; survivability; de Bruijn sequences.*