**Національний технічний університет України**

**«Київський політехнічний інститут»**

факультет інформатики та обчислювальної техніки

кафедра обчислювальної техніки

Курсовий проект

*з дисципліни:* **«Проектування комп’ютерних систем»**

|  |  |
| --- | --- |
| **Виконав**: | Мазур Роман Федорович |
| **Група:** | ІО-52 |
| **Перевірили**: | Луцький Г. М. \_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис керівника)  Кулаков Ю. А. \_\_\_\_\_\_\_\_\_\_\_\_\_  (підпис керівника) |

2009 р.

ТЕХНІЧНЕ ЗАВДАННЯ

1. Синтезувати оригінальну масштабовану топологічну організацію. Представити рисунок організації.
2. Визначити топологічні характеристики синтезованої організації. Проаналізувати такі параметри, як: ступінь, діаметр, середній діаметр, вартість, трафік, зв’язність.
3. Порівняти синтезовану топологію з гіперкубом.
4. Відобразити топологію в гіперкуб.
5. Розробити відмовостійкі алгоритми для чотирьох видів маршрутизації для синтезованої топології:
   1. один до одного,
   2. усі до всіх з персональним призначенням,
   3. один до всіх,
   4. multicast.
6. Розглянути розв’язання прикладної задачі на синтезованій топологічній організації.

ПОЯСНЮВАЛЬНА ЗАПИСКА

**Зміст**

[1 Синтез топології 6](#_Toc250904874)

[1.1 Принципи побудови 6](#_Toc250904875)

[1.2 Розмір та нумерація топології 8](#_Toc250904876)

[1.3 Закони формування зв’язків у топології 9](#_Toc250904877)

[1.4 Базові функції для роботи з топологією 9](#_Toc250904878)

[2 Аналіз топології 11](#_Toc250904879)

[2.1 Визначення топологічних характеристик 11](#_Toc250904880)

[2.2 Порівняння топології з гіперкубом 12](#_Toc250904881)

[3 Відображення топології у гіперкуб 13](#_Toc250904882)

[4 Маршрутизація 14](#_Toc250904883)

[4.1 Один до одного 14](#_Toc250904884)

[4.2 Усі до всіх з персональним призначенням 14](#_Toc250904885)

[4.3 Один до всіх 14](#_Toc250904886)

[4.4 Multicast 14](#_Toc250904887)

[5 Завантаження прикладної задачі 15](#_Toc250904888)

1 Синтез топології

1.1 Принципи побудови

В основу структурної організації топології було вирішено покласти гіперкуб у зв’язку з тим, що його побудова дає можливість завантажувати дуже великий спектр прикладних задач. Для того, щоб збільшити відмовостійкість топології основою його системи числення була вибрана тріскова замість двійкової, класичної. Цікавою особливістю такої топології є те, що трійковий гіперкуб 2-го порядку вироджується у меш-топологію 3х3 (рис. 1.1).



Рис. 1.1 Трійковий гіперкуб 2-го порядку

Основною вадою гіперкубічних топологій є їхній високий степінь, який визначається алгебраїчно степеневою функцією (*аN*). У нашому випадку ріст значення степеня буде стрімкішим за класичний через більше значення основи системи числення. Це, у свою чергу, спричиняє швидке зростання вартості системи. Таким чином, необхідно зупинити зростання степеня топології.

Для мінімізації вартості системи вершиною гіперкуба виступає не окремий вузол, а кластер з кількістю вузлів рівною подвоєному значенню порядку системи. Оскільки, в такому випадку кількість вузлів у кластері буде рівною степеню вершини 3-вого гіперкуба, то кожен з вузлів матиме один зв’язок з іншим кластером. Найпростіший варіант зв’язків у кластері – кільце (рис. 1.2).

|  |  |
| --- | --- |
| а) | б) |
| Рисунок 1.2 а) вершина трійкового гіперкуба 3-го порядку,  б) зменшення степеня вершини | |

Зі збільшенням порядку топології все більше значення матимуть параметри зв’язності внутрішнього кластера, а не зовнішнього гіперкуба. Тому, щоб збільшити зв’язність у кластері, введено додаткове ребро з «протилежною» вершиною. У результаті отримано топологічну організацію, показану на рисунках 1.3 та 1.4.

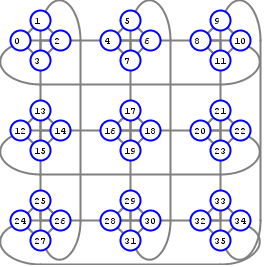


Рисунок 1.3 Топологічна організація 2-го порядку

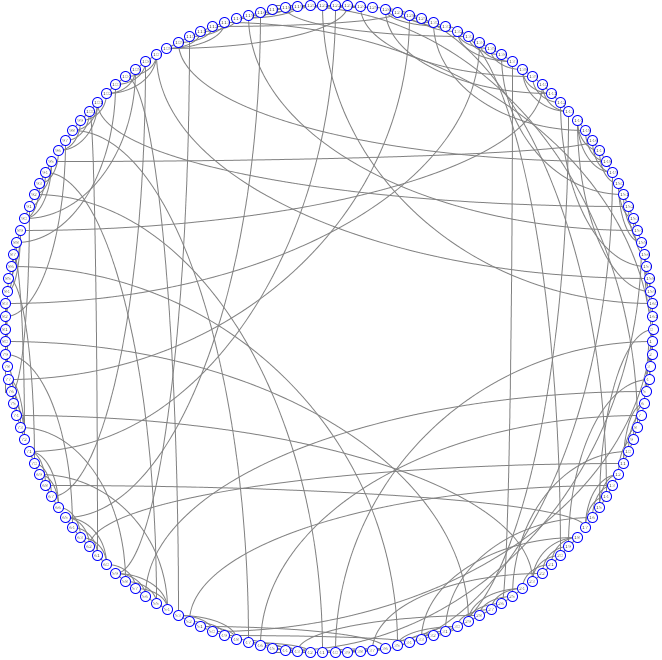


Рисунок 1.4 Топологічна організація 3-го порядку

1.2 Розмір та нумерація топології

Топологічна організація формується з кластерів. Їхня кількість визначається кількістю вершин трійкового гіперкуба відповідного порядку. Тобто, для організації з порядком *d* матимемо 3*d* кластерів. Кожен кластер має розмір (кількість вузлів, що його формують) рівний степеню вершини трійкового гіперкуба, тобто .

Кожен вузол у топології має наскрізний номер, однак в алгоритмах, звісно, часто використовуються поняття номера кластера та номера вузла у кластері. Перший може бути визначений, як частка від ділення наскрізного номера на розмір кластера, а другий, як остача цього ділення.

1.3 Закони формування зв’язків у топології

Як уже було сказано, кожен вузол у синтезованій топології має 4 зв’язки: три з них – зв’язки всередині кластера, і ще один – ребро зовнішнього трійкового гіперкуба. Нехай, *d* – порядок топології, *cs* – розмір кластера, *і* – номер вузла (наскрізний), *N* – кількість вершин у топології. Тоді чотири зв’язки *і*-го вузла можуть бути описані наступним чином ().

1. – зв’язок усередині кластера, за годинниковою стрілкою, рисунок 1.5 а).
2. – зв’язок усередині кластера, проти годинникової стрілки, рисунок 1.5 б).
3. – зв’язок усередині кластера з «протилежною» вершиною, рисунок 1.5 в).
4. – зв’язок з відповідною вершиною сусіднього кластера, рисунок 1.5 г).



Рисунок 1.5 Чотири зв’язки між вузлами

1.4 Базові функції для роботи з топологією

Далі наведемо основні функції, що можуть бути корисними при роботі з синтезованою топологічною організацією. В усіх формулах *d* – порядок топології, *і* та *j* – номера вузлів.

|  |  |
| --- | --- |
| Таблиця 1.1 Базові функції | |
| Кількість вузлів |  |
| Розмір кластера |  |
| Номер кластера |  |
| Номер вузла у кластері |  |
| Номер відповідної вершини сусіднього кластера | **int** axis = ni % d;  **int** sourceConnector = ni / d;  **int** nearDigit = *getDigit*(ci, axis);  nearDigit += sourceConnector == 0 ? -1 : 1;  nearDigit %= 3;  **if** (nearDigit < 0) { nearDigit=3+nearDigit; }  **int** nearCluster = *setDigit*(ci,axis,nearDigit);  **return** nearCluster \* cs + (ni + d) % cs; |
| Отримання масиву сусідніх вершин | **int** startNode = i / cs \* cs;  **return** **new** Integer[] {  startNode + (i + 1) % cs,  startNode + (i > 0 ? (i - 1) % cs : cs - 1),  startNode + (i + d) % cs,  *getNearClusterConnection*(i, d)  }; |

2 Аналіз топології

2.1 Визначення топологічних характеристик

Розглянемо наступні топологічні характеристики:

* N – кількість вузлів;
* S – степінь;
* D – діаметр;
* Dсер – середній діаметр;
* C – вартість;
* T – трафік.

Дані параметри залежатимуть від порядку топологічної організації, тому розглянемо закони, за якими вони можуть бути визначені.

Кількість вузлів має дуже стрімке зростання при масштабуванні та визначається за формулою

Степінь є геометричною характеристикою, що характеризує локальну зв’язність організації та визначається як максимальне число ребер, інцидентних одній вершині. Для синтезованої топологічної організації степінь є константою, що було однією із задач синтезу.

Діаметр топології є мінімальною відстанню між максимально віддаленими вершинами в топології. Його можна визначити через матрицю зв’язності. Для синтезованої організації діаметр визначається формулою

Середній діаметр показує усереднену відстань між вузлами у топології та обчислюється за допомогою матриці відстаней ().



Вартість визначається як мультиплікативний критерій діаметра, кількості вузлів та степеню.

Вартість показує, на скільки раціонально використовувати топологію при заданих параметрах, тобто фактично визначає співвідношення «ціна-якість».

Трафік (або густина трафіку) визначається за формулою

Дана характеристика показує ефективність використання ребер у топології. Чим ближче її значення до 1, тим ефективнішим є потенційне використання. У таблиці 2.1 наведено дані показники для різного порядку топології.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Таблиця 2.1 Топологічні характеристики | | | | | | |
| d | N | S | D | Dсер | T | C |
| 2 | 36 | 4 | 5 | 3,11 | 2,08 | 540 |
| 3 | 162 | 4 | 9 | 5,27 | 3,52 | 4374 |
| 4 | 648 | 4 | 11 | 6,99 | 4,66 | 21384 |

2.2 Порівняння топології з гіперкубом

Текст

3 Відображення топології у гіперкуб

Для відображення синтезованої топологічної організації у гіперкуб природно представити кожен з кластерів як одну вершину гіперкуба. У результаті отримаємо трійковий гіперкуб (рисунок 3.1). Таким чином задача відображення зводиться до відображення гіперкуба з системою числення 3 до двійкового.

|  |  |
| --- | --- |
|  |  |

Рисунок 3.1 Перехід до трійкового гіперкуба

4 Маршрутизація

4.1 Один до одного

4.2 Усі до всіх з персональним призначенням

4.3 Один до всіх

4.4 Multicast

5 Завантаження прикладної задачі