

**Д.т.н, професор Романкевич О.М., магістрант Буланов О.М.**

**Національний технічний університету України  
«Київський політехнічний інститут»**

**СПОСІБ ОРГАНІЗАЦІЇ РЕКОНФІГУРУВАННЯ БАГАТО-  
ПРОЦЕСОРНИХ СИСТЕМ З УРАХУВАННЯМ ОБМЕЖЕНЬ ПО  
ПРОДУКТИВНОСТІ**

**Abstract**

*Oleksiy M. Romankevuch, prof., DcS; Oleksandr Bulanov, student  
The way to organize reconfiguration of multiprocessor systems given the  
constraints on productivity*

*The problems of organization reconfiguration in multiprocessor control systems at various stages of their design and operation are considering. An algorithm for reduction of brute force in determining the placement of management functions to the system processor, which is based on a table, which is transformed as they become failures of processors. This algorithm is implemented as a software emulator, which demonstrates the reconfiguration process of multiprocessor systems.*

**Вступ**

Відмовостійкі багатопроцесорні системи з кожним роком знаходять все більш широке застосування не тільки в обчислювальних структурах, а й у системах управління складними об'єктами. Крім високого ступеня універсальності, вони мають ряд переваг, зокрема, в плані підвищення головних параметрів систем обробки інформації: швидкодії та надійності. Такі системи мають структурну і тимчасову надлишковість, здатні самі себе тестувати, виявляти і виключати з роботи процесори, що вийшли з ладу, реконфігуруватися та продовжувати роботу в повному обсязі.

**Постановка задачі**

Задача полягає в розробленні алгоритму організації реконфігурування багатопроцесорних систем з урахуванням обмежень по продуктивності та впровадженні його до програми емулятора, з метою підвищення відмовостійкості такої системи.

## Аналіз задачі

Будь-яка система управління деяким об'єктом виконує певну множину функцій управління  $F = f_1, f_2, \dots, f_N$ . Певну підмножину  $F_{on} \subset F$  цих функцій, невиконання хоча б однієї з яких викликає появу небезпечного стану об'єкта управління, називають небезпечною, а функції з  $F_{on}$  - небезпечними. Бувають об'єкти, що не мають небезпечних станів - безпечні об'єкти. Існують системи (об'єкт плюс система управління), які перебувають у повністю непрацездатному стані, не будучи в небезпечному. Деяка множина  $F_{oon}$  безпечних функцій, які система управління може при цьому виконувати, в цій роботі виключимо з розгляду, оскільки для завдання, що розглядається, це не має значення.

Для уточнення понять, якими оперуємо, допустимо наступне. Згідно з основними положеннями теорії гарантоздатності, можна зазначити, що система може перебувати в одному з декількох станів: працездатний (виконуються всі функції з  $F$ ), частково працездатний (виконуються функції з деякого підмножини функцій  $F' \subset F$ , але  $F_{on} \subset F'$ ), повністю непрацездатний (може бути небезпечним і безпечним).

Серед особливостей відмовостійких багатопроцесорних систем особливе значення має можливість реконфігурування: після виконання процедури взаємного тестування процесорів і встановлення процесора, що вийшов з ладу, відбувається перерозподіл його функцій серед процесорів, що залишилися працездатними. Критерії оптимізації процесу реконфігурації залежать від типу відмовостійкої багатопроцесорної системи. Серед них можна відзначити наступні: забезпечення максимальної надійності працездатної частини відмовостійкої багатопроцесорної системи, що залишилася, сумарний час, необхідний для виконання своєї і чужої «функції» або відповідний обсяг пам'яті, мінімізація ймовірності настання небезпечного стану при подальшій відмові та ін.

Слід зазначити, що в багатопроцесорних системах управління досить часто мають місце різні обмеження на розподіл функцій по процесорах. Зрозуміло, що кожна функція для своєї реалізації потребує кількості операцій в одиницю часу, тобто частини продуктивності процесора, що можна позначити як  $s_f$ . Якщо  $s_p$  - продуктивність процесора, то має місце, цілком очевидне, обмеження -  $s_p \geq \sum s_{f_i}$  для множини функцій, які потрапляють в даний процесор.

Друге обмеження: кожна функція може бути виконана в одному з процесорів деякої підмножини процесорів системи. Це обмеження може бути викликано різними причинами: особливості та можливості процесора і його програмного забезпечення, наявність або відсутність каналів зв'язку з джерелами інформації ззовні та ін.

Зазначені обмеження серйозно позначаються на можливостях реконфігурування відмовостійких багатопроцесорних систем, особливо тоді, коли початкова надлишковість зникає (в міру появи відмов процесорів), і від виконання деяких функцій доводиться відмовлятися.

В останньому випадку система перестає бути відмовостійкою і переходить в режим часткової працездатності, втрачаючи деякі функції управління. Якщо вони еквівалентні, то особливих проблем при виборі множини функцій, які все-таки повинні виконуватися, не виникає. Якщо ні, то критерієм може бути їх кількість. Це було б дуже зручно, так як відкидання найбільш вагомих функцій після кожної відмови в першу чергу призводило б до появи певної надлишковості по продуктивності, яка може бути використана з різною метою.

## Алгоритм

Поки число можливих реальних відмов менше ступеня відмовостійкості системи існує досить багато рішень, тому що є певна надлишковість. Потім, у міру появи відмов доводиться вирішувати завдання оптимізації процесу реконфігурування згідно з прийнятим критерієм при недопущенні (поки це можливо) попадання системи в небезпечний стан.

Нехай є багатопроцесорна система, яка містить  $n$  процесорів, причому система повинна виконувати  $N$  функцій управління, і є зазначені обмеження. У прикладі, розглянутому нижче,  $n=6$  і  $N=10$ , причому функція  $f_i$  може бути виконана або процесором  $P_{i1}$ , або процесором  $P_{i2}$ , функція  $f_j$  - одним із процесорів  $P_{j1}$ ,  $P_{j2}$ ,  $P_{j3}$  або  $P_{j6}$  і т.д., відповідно до табл. 1. Кожен процесор має свою продуктивність  $s_{pi}$ , кожна функція вимагає для свого виконання -  $s_{fi}$ .

Алгоритм починається з побудови таблиці, подібної табл. 1. В ній рядки позначені  $N_i$  - підмножини функцій, які здатний виконувати  $i$ -й процесор, стовпчики вказують на підмножину процесорів, куди функція  $f_i$  може бути поміщена. Для простоти вважаємо, що всі процесори мають однакову

продуктивність (хоча це не обов'язково), що дорівнює 6 умовних одиниць. Нижня рядок таблиці - продуктивність (у тих же умовних одиницях), яка потрібна для виконання відповідної функції.

Далі визначається  $s' = \sum s_{f_i}$  для функцій, що підлягають виконанню, а також  $s = \sum s_{p_i}$  для процесорів, що є працездатними. Для відмовостійких багатопроцесорних систем до появи відмов, розподіл функцій не складає труднощів, як уже зазначалося раніше, оскільки є надлишковість, і  $s > s'$ .

Таблиця 1

Розподілення функцій

	f <sub>1</sub>	f <sub>2</sub>	f <sub>3</sub>	f <sub>4</sub>	f <sub>5</sub>	f <sub>6</sub>	f <sub>7</sub>	f <sub>8</sub>	f <sub>9</sub>	f <sub>10</sub>
N <sub>1</sub>	*	*		*	*		*			
N <sub>2</sub>		*			*	*			*	*
N <sub>3</sub>	*					*		*	*	
N <sub>4</sub>			*	*			*			*
N <sub>5</sub>		*	*			*	*			
N <sub>6</sub>		*			*		*	*		*
	3	2	3	4	2	3	3	3	3	4

Далі, після появи чергової відмови будується нова таблиця (без відповідного рядка), і по ній визначається новий розподіл функцій. Деяке полегшення у виборі з'являється, якщо в стовпці є єдина позначка.

Якщо розподіл виявляється неможливим, наприклад, коли  $s' > s$ , то доводиться виключати з множини F деяку підмножину  $\sigma$  функцій відповідно до обраного критерію. При цьому  $\sigma \cap F_{on} = \emptyset$ . Будується нова таблиця шляхом викреслювання стовпчиків, відповідних функцій з  $\sigma$ , для якої знову підраховуються  $s$  і  $s'$ , і проводиться пошук розподілу, тобто вибираються такі комбінації по рядках щоб кожна функція брала участь лише один раз, та сума ваги функцій, які залишились в кожному рядку не перевищувала продуктивності відповідного процесора.

Якщо пошук не дає результату, доводиться вибрати іншу підмножину  $\sigma$ , будувати іншу таблицю, і процес повторюється. Те ж саме відбувається при виявленні кожної наступної відмови. В міру появи відмов процесорів таблиця зменшується, і перебір варіантів вибору  $\sigma$  і розподілу функцій скорочується. Кінець настає тоді, коли умова  $\sigma \cap F_{on} = \emptyset$  стає нездійсненною.

## **Емулятор**

В рамках даної роботи було розроблено емулятор відмовостійкої багато-процесорної системи, в якій відбувається оптимальне реконфігурування функцій та процесорів після виникнення відмови процесора, враховуючи обмеження по продуктивності за вищеописаним алгоритмом. Слід зазначити що програма виконує реконфігурування для будь-якого числа функцій та процесорів, а також з можливістю встановлення індивідуальної частоти для кожного окремого процесора.

## **Висновок**

Вище детально проаналізовано алгоритм організації реконфігурування багатопроцесорної системи з урахуванням обмежень по продуктивності. Даний алгоритм було реалізовано у вигляді програми-емюлятора багато-процесорної системи для підвищення її відмовостійкості.

Слід зазначити що даний алгоритм не є досконалим оскільки тут завжди має місце деякий перебір що, залишає можливість вдосконалення та підвищення швидкодії для реконфігурації.

## **Література**

1. *Каляев И.А., Левин И.И., Семерников Е.А.*, Реконфигурируемые мультиконвейерные вычислительные структуры – Ростов-на-Дону: издательство ООО «ЮНЦ РАН», 2008, – С. 20-140.
2. *Додонов А.Г., Ландэ Д.В.*, Живучесть информационных систем — Киев: издательство «Наукова думка», 2011, - С. 50-190.