

# МЕТОДИ РОЗПАРАЛЕЛЮВАННЯ ВИРШУВАЧА РІВНЯНЬ MIMD-МОДЕЛІ МЕРЕЖНИХ ДИНАМІЧНИХ ОБ'ЄКТІВ

Святний В.А., Смагін О.М., Солонін О.М.  
кафедра ЕОМ ДонНТУ,  
[solonin@cs.dgtu.donetsk.ua](mailto:solonin@cs.dgtu.donetsk.ua)

## Abstract

*Svjatnij V., Smagin O., Solonin O. Methods of parallelisation for the equations solver of MIMD – model of net dynamic object. Various approaches to processes of devirtualisation of MIMD - models of net objects are considered. Two new methods of distribution of virtual processes of MIMD - model between accessible processors are described.*

## 1. Вступ

Мережні динамічні об'єкти (МДО) різної фізичної природи відносяться до складних динамічних систем. Формальний опис МДО є основою для побудови математичних моделей і складається з опису топології і рівнянь динамічних процесів [1, 2].

Топології мережних об'єктів представляються орієнтованими графами  $G(U, V)$ , що кодуються наступною таблицею:

$$NUJ \ KIJ \ QI \ PAR \ KOM, \quad (1)$$

де NUJ, KIJ - номери початкових і кінцевих вузлів орієнтованих гілок; QI – номери гілок, PAR – параметри гілок, KOM – коментарі;  $|V|=m$  – кількість гілок,  $|U|=n$  – кількість вузлів.

Для мережних об'єктів із зосередженими параметрами вузлові рівняння нерозривності мають вигляд:

$$AQ=0, \quad (2)$$

де A – матриця інциденцій, Q - вектор потоків повітря.

Контурні рівняння для вентиляційної мережі довільної топології записуються так:

$$SK \frac{dQ}{dt} + SRZ = SH . \quad (3)$$

Тут S – матриця контурів; R, K – діагональні матриці аеродинамічних параметрів, Z – вектор з елементами типу  $Q_i|Q_i|$ ; H – вектор тисків вентиляторів. Елементи цих векторів можуть бути нелінійними функціями  $H_i(Q_i)$ .

Моделююче програмне забезпечення, що реалізує моделі мережних об'єктів, розділяють на три послідовні частини [2]:

- Топологічний аналізатор (ТА). Основним результатом роботи ТА є дерево та антидерево графа, матриця інциденцій  $A$  та матриця контурів  $S$ , що зв'язують топологічний опис з рівняннями процесів.
- Генератор рівнянь (ГР), що приводить систему (2), (3) у вигляд, зручний для чисельного вирішення [1, 2]:

$$\begin{cases} \dot{Y} = Hu - Ru * Z \\ X = -W * Y \\ W = Ax^{-1} * Ay \end{cases} \quad (4)$$

$$\begin{aligned} Hu &= (Sy * Ky - Sx * Kx * W)^{-1} * S * H = TP * H \\ Ru &= (Sy * Ky - Sx * Kx * W)^{-1} * S * R = TP * R \end{aligned} \quad (5)$$

- Вирішувач рівнянь (ВР), що реалізує обраний алгоритм чисельного пошуку невідомих векторів  $X$ ,  $Y$  системи (4), (5).

З огляду на складність мережних динамічних об'єктів постала проблема побудови паралельних ВР [2, 3]. В рамках міжнародного наукового співробітництва ДонНТУ та Штутгартського університету набуто певний досвід побудови SIMD-вирішувачів [2], скомпоновано досліди зразок паралельного моделюючого середовища [4]. Ресурси середовища дозволяють провести комплексну розробку та дослідження паралельного MIMD-вирішувача системи рівнянь (4), (5). Цій проблемі не приділялось достатньої уваги [2, 3].

Метою даної статті є побудова вирішувача рівнянь паралельної моделі мережних динамічних об'єктів на основі запропонованих методів розпаралелювання віртуальних моделей та їх девіртуалізації і відображення на наявні паралельні обчислювальні системи.

## 2. Віртуальна паралельна модель мережного об'єкта

На рис.1 приведено граф тестового мережного об'єкта ( $m=117$ ,  $n=61$ ). Блок-схема послідовного алгоритму функціонування вирішувача рівнянь (4) – (5) представлена на рис.2. Аналізуючи можливості розпаралелювання операцій по обчисленню компонентів векторів  $X$ ,  $Y$ , можна запропонувати ряд варіантів віртуальної MIMD – моделі мережних об'єктів. При цьому під віртуальною будемо розуміти модель, у якій виділені паралельно функціонуючі відносно автономні процеси, що спільно можуть забезпечити вирішення задачі (4), (5).

Схему віртуальної моделі, що розпаралелена за SPMD-принципом відповідно до кількості гілок, приведено на рис. 3. У цій моделі передбачено  $m$  віртуальних процесів, кожний з яких веде обчислення з визначеними рядками матриць  $X$ ,  $Y$  і  $W$ . Після обчислення поточних значень векторів  $Y$  відбувається обмін інформацією між процесами, далі компонується вектор  $Q = (X, Y)^T$  і знову відбувається обмін. Порядок обчислень і обмінів ведеться відповідно до системи (4), (5).

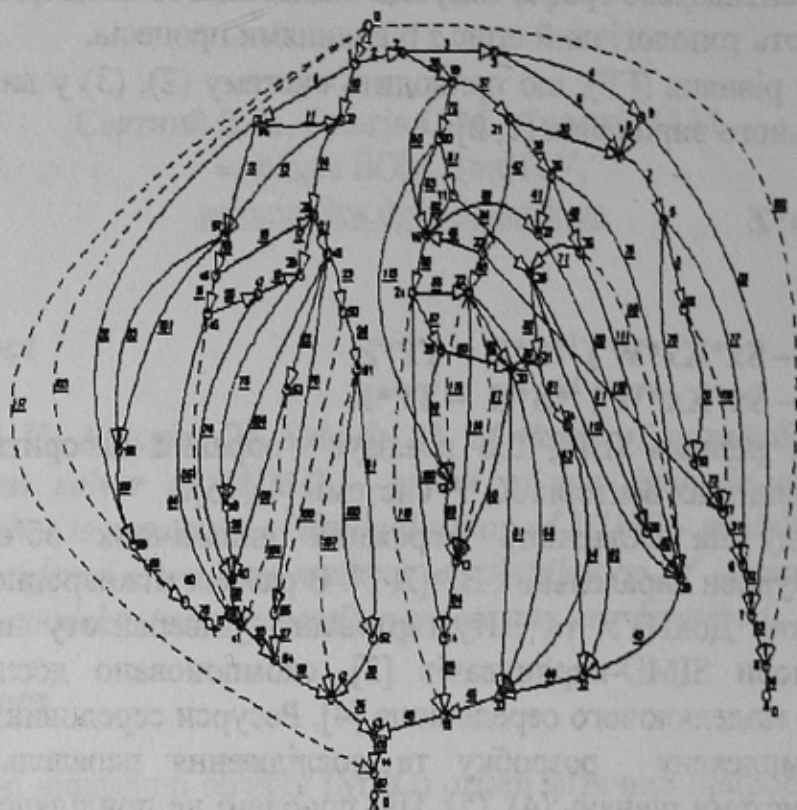


Рис. 1 – Граф тестового мережного об'єкта

### **3. Девіртуалізація віртуальних паралельних моделей**

Девіртуалізацією назвемо процес відображення віртуальної моделі (рис. 3) на цільову паралельну обчислювальну систему (ЦПОС), тобто ту обчислювальну систему, що є в розпорядженні користувача.

Пропонуються наступні етапи девіртуалізації:

- 1) Априорна оцінка первісної віртуальної моделі і розгляд варіантів віртуальної моделі, що мають кращі (якісно) показники за рівномірністю завантаження процесів та обсягами обмінюваних даних.

Аналізуючи схему рис.3, можна запропонувати наступні варіанти підходів до розпаралелювання:

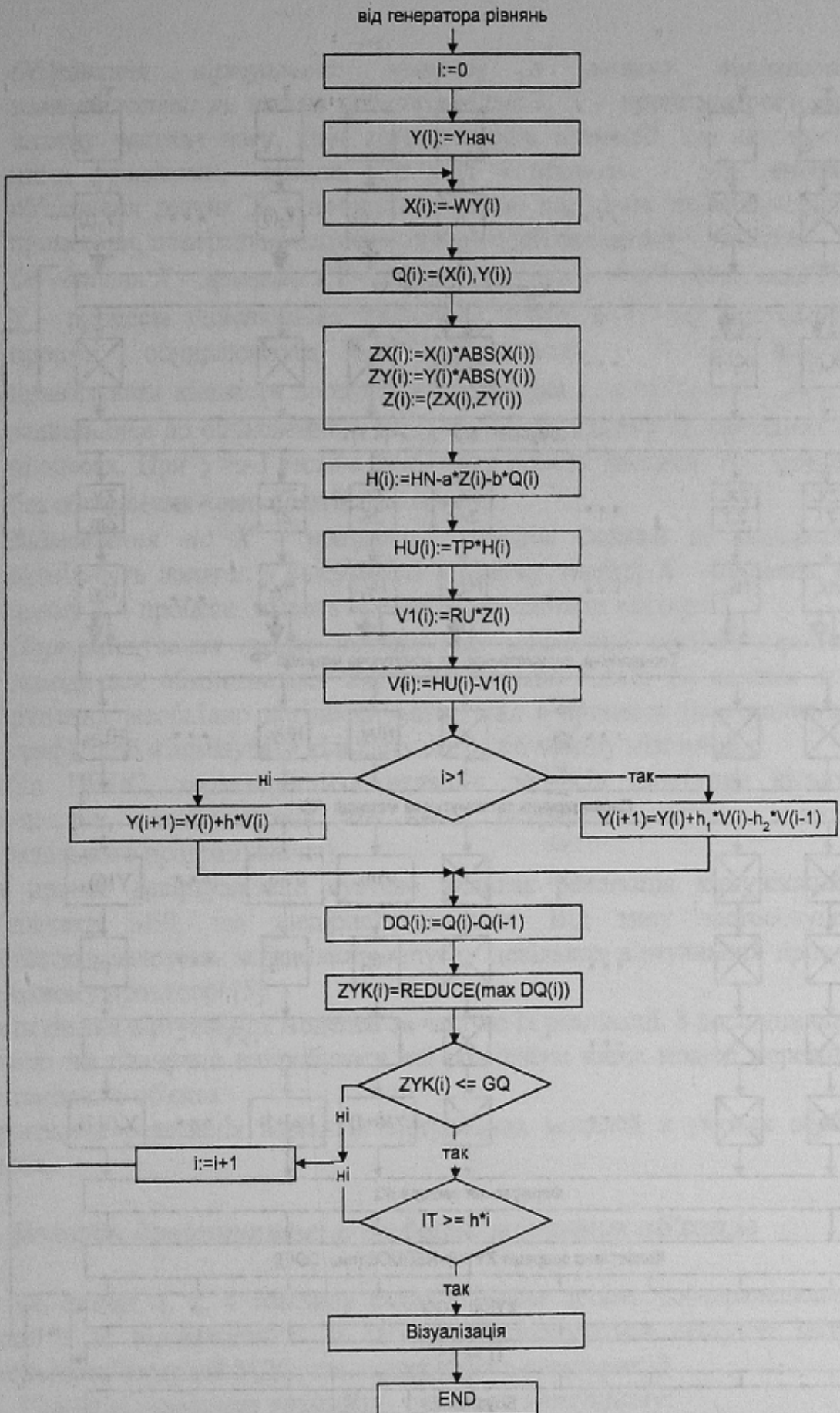


Рис. 2 - Блок-схема алгоритму функціонування вирішувача рівнянь

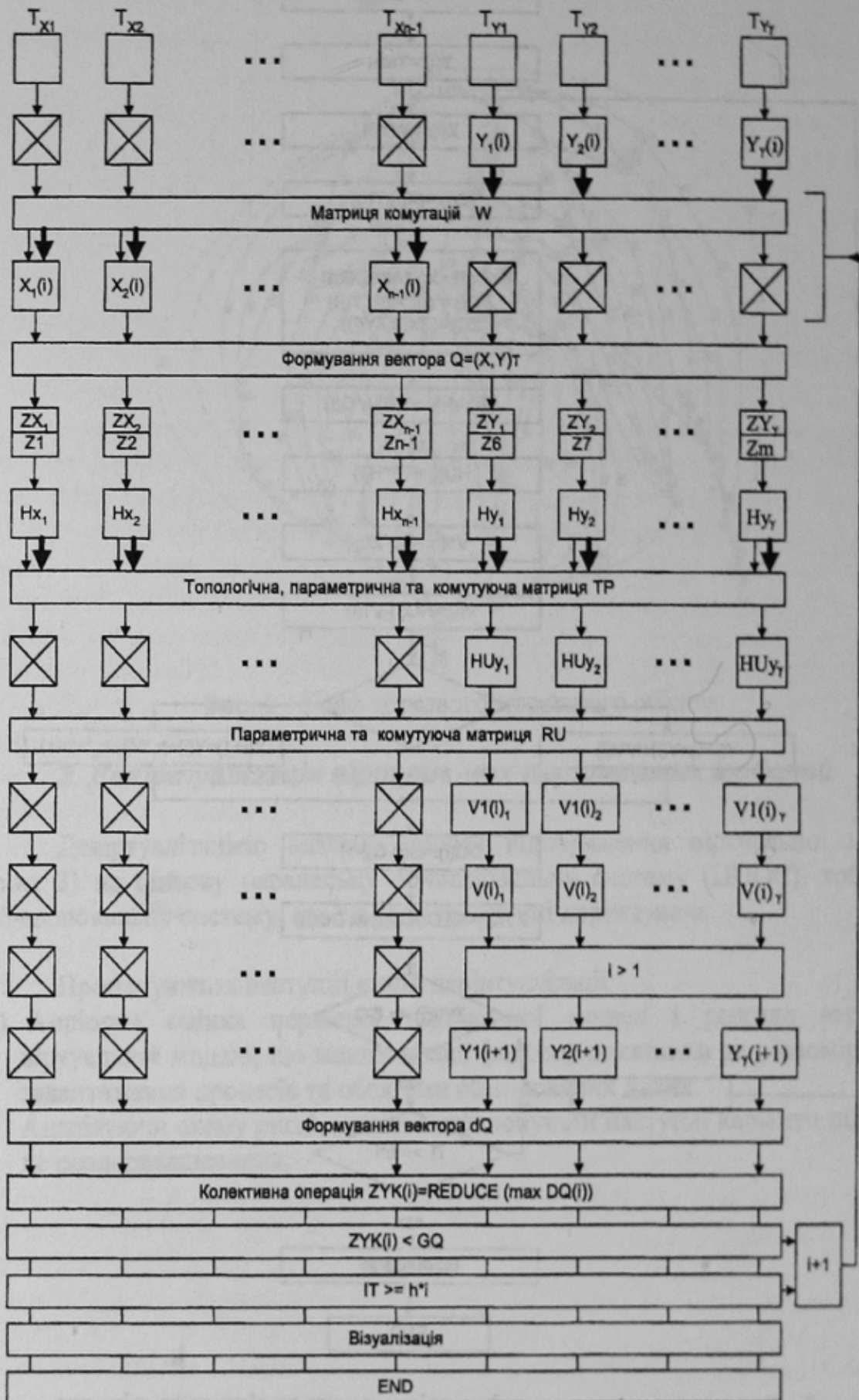


Рис. 3 - Схема розпаралелювання програми за SPMD-принципом

- *Об'єднання віртуальних процесів з метою вирівнювання навантаження:* як можна побачити з рис.3,  $X$  – процеси простоюють істотну частину часу, крім того кількість операцій, що виконується цими процесами, менша, ніж у  $Y$  – процесів. Є сенс виконати об'єднання деяких  $X$  – процесів з метою вирівняти навантаження на процесори, попередньо визначивши критерії складності процесів;
  - *Об'єднання  $X$  – процесів з  $Y$  – процесами:* при  $\gamma = n-1$  об'єднання  $X$  – і  $Y$  – процесів здійснюється природньо, тобто в  $i$  – му віртуальному процесі обчислюються  $X_i, Y_i$ . У ситуації  $\gamma < n-1$  доцільно прив'язувати кількість процесів до величини  $\gamma$ , а ті  $X_{\gamma+1}, \dots, X_{n-1}$ , що залишилися до обчислення, концентрувати в одному чи декількох  $X$  – процессах. При  $\gamma > n-1$  можна допустити роботу частини  $Y$  – процесів без обчислення компонентів вектора  $X$ .
  - *Відмовлення від  $X$  – процесів:* випадок схожий на попередній, відмінність полягає у відсутності в явному вигляді  $X$  – процесів, при цьому  $Y$  – процеси містять масиви зі значеннями вектора  $X$ .
  - *Партиціонування графа мережі при мінімізації обмінів:* процесам даводиться обмінюватися інформацією між собою на кожному кроці рішення, необхідно так поєднувати гілки в процессах (партиціонувати граф), щоб мінімізувати кількість операцій обміну між ними.
- 2) Вибір ЦПОС, характеристика наявних ресурсів (доступна кількість процесорів, схема зв'язків між ними, операційна система і засоби паралельного програмування).
- На процес девіртуалізації суттєво впливає реалізація комунікаційної бібліотеки MPI, що використовувається. Від типу застосовуваної бібліотеки залежить можливість запуску декількох віртуальних процесів на одному процесорі [5].
- 3) Ранжування віртуальних моделей за чергою їх реалізації. З дослідницькою метою ми плануємо випробувати всі розглянуті вище моделі мережного динамічного об'єкта.
- 4) Програмна реалізація варіантів віртуальних моделей в умовах обраної ЦПОС.

#### **4. Методи девіртуалізації моделей мережних об'єктів**

На етапах 1, 2, 4 важливо формалізувати процес розпаралелювання моделей і їх відображення на ЦПОС. Пропонуються наступні методи девіртуалізації моделей МДО, що модифікують схему рис. 3.

Метод рівномірного розподілу полягає в наступному:

Нехай  $m$  – кількість гілок МО,  $n$  – кількість вершин,  $(n-1)$  – кількість гілок дерева,  $\gamma = m - (n-1)$  – цикломатичне число (кількість незалежних контурів), що дорівнює кількості гілок антидерева,  $P$  – кількість віртуальних

процесів ( $P = m = \gamma + (n-1)$ ),  $N_{CPU}$  - кількість доступних процесорів ( $1 \leq N_{CPU} \leq P$ ).

Необхідно рівномірно розподілити  $P$  віртуальних процесів ( $(n-1)$  і  $\gamma$ ) між  $N_{CPU}$  доступними процесорами, тобто створити розпаралелювання, яке можна масштабувати. Будемо розподіляти віртуальні процеси відповідно гілкам дерева  $P_1 = n-1$  і антидерева  $P_2 = \gamma$  (тут  $P = P_1 + P_2$ ) між двома групами процесорів  $N_1$  і  $N_2$  відповідно, де  $N_{CPU} = N_1 + N_2$ .

Позначимо  $\text{quot}[F]$  – цілу частину числа  $F$ , а  $\text{rem}[F]$  – дробову частину, тоді:

$$N_1 = \text{quot} \left[ \frac{P_1 \cdot N_{CPU}}{P} \right]; \quad N_2 = \begin{cases} \text{quot} \left[ \frac{P_2 \cdot N_{CPU}}{P} \right]; \text{rem} \left[ \frac{P_2 \cdot N_{CPU}}{P} \right] < 0,5 \\ \text{quot} \left[ \frac{P_2 \cdot N_{CPU}}{P} \right] + 1; \text{rem} \left[ \frac{P_2 \cdot N_{CPU}}{P} \right] \geq 0,5 \end{cases};$$

$P_1$  можна представити формулою:

$$P_1 = F_1 * N_1 + E_1; (E_1 < N_1; N_1 = D_1 + E_1).$$

Розподілимо  $P_1$  віртуальних процесів між  $N_1$  процесорами у такий спосіб:

на  $D_1$  процесорах –  $F_1$  віртуальних процесів,

на  $E_1$  процесорах –  $(F_1 + 1)$  віртуальних процесів, тобто

$$D_1 * F_1 + E_1 * (F_1 + 1) = n - 1 = P_1.$$

$P_2$  можна представити як :  $P_2 = F_2 * N_2 + E_2; (E_2 < N_2; N_2 = D_2 + E_2).$

Розташуємо  $P_2$  віртуальних процесів на  $N_2$  процесорах подібно до розподілу  $P_1$ :

на  $D_2$  процесорах –  $F_2$  віртуальних процесів,

на  $E_2$  процесорах –  $(F_2 + 1)$  віртуальних процесів, тобто

$$D_2 * F_2 + E_2 * (F_2 + 1) = \gamma = P_2.$$

Таким чином, розподіл процесів віртуальної MIMD – моделі на процесори ЦПОС визначається параметрами  $m, n, \gamma, D_1, D_2, E_1, E_2, F_1, F_2, P_1, P_2$ .

Метод рівномірного розподілу по гілках антидерева являє собою модифікований метод рівномірного розподілу, розглянутий вище.

Як можна побачити з рис.3, віртуальні процеси  $X_i$  велику частину роботи програми простоюють, а при обчисленнях істотний час витрачається на обмін за матрицею комутації  $W$ . Сутність методу рівномірного розподілу по гілках антидерева полягає у відсутності в явній формі групи віртуальних  $X$  – процесів. У програмі присутні тільки  $Y$  – процеси, кожний з яких містить свою копію даних, відповідних  $X$  – процесам, тобто обчислення вектора  $Y$  розподілено між  $\gamma$  віртуальними процесами

$$\begin{pmatrix} Y_1 \\ Y_2 \\ \dots \\ Y_\gamma \end{pmatrix} = \begin{pmatrix} Hu_1 - Ru_{11} * Z_1 + Ru_{12} * Z_2 + \dots + Ru_{1m} * Z_m \\ Hu_2 - Ru_{21} * Z_1 + Ru_{22} * Z_2 + \dots + Ru_{2m} * Z_m \\ \dots \\ Hu_\gamma - Ru_{\gamma 1} * Z_1 + Ru_{\gamma 2} * Z_2 + \dots + Ru_{\gamma m} * Z_m \end{pmatrix}, \quad (6)$$

кожний з яких обчислює свою складову вектора X ( $X = -W * Y$ )

$$X = \begin{pmatrix} W_{11}Y_1 + W_{12}Y_2 + \dots + W_{1\gamma}Y_\gamma \\ W_{21}Y_1 + W_{22}Y_2 + \dots + W_{2\gamma}Y_\gamma \\ \dots \\ W_{n-1,1}Y_1 + W_{n-1,2}Y_2 + \dots + W_{n-1,\gamma}Y_\gamma \end{pmatrix}, \quad (7)$$

тобто у віртуальному процесі  $Y_i$ ; знаходиться масив доданків вектора X:

$$Y_i : \{X_{1(Y_i)} = W_{1\gamma} * Y_\gamma; X_{2(Y_i)} = W_{2\gamma} * Y_\gamma; \dots; X_{n-1(Y_i)} = W_{n-1,\gamma} * Y_\gamma;\} \quad (8)$$

Після обчислення вектора Y кожен Y – процес обчислює свою частину вектора X, а потім виконується глобальна операція редукції з підсумовуванням.

Нехай P - кількість віртуальних процесів ( $P = \gamma$ ),  $N_{CPU}$  - кількість доступних процесорів ( $1 \leq N_{CPU} \leq P$ ). Тоді P можна представити формулою:

$$P = F * N_{CPU} + E; (E < N_{CPU}; N_{CPU} = D + E).$$

Розподілимо P віртуальних процесів між  $N_{CPU}$  процесорами у такий спосіб:

на D процесорах – F віртуальних процесів,

на E процесорах – (F+1) віртуальних процесів, тобто

$$D * F + E * (F + 1) = \gamma = P.$$

Таким чином, розподіл процесів віртуальної MIMD – моделі на процесори ЦПВС визначається параметрами m, n,  $\gamma$ , D, E, F<sub>2</sub>, P.

## 5. Імплементация та порівняльний аналіз моделей

Модель мережного об'єкта з  $m=117$ ,  $n=61$  (рис.1) реалізована авторами на ЦПОС CRAY T3E обчислювального центру Штутгартського університету [4]. У моделі використано 117 процесорів за схемою рис.3, час моделювання склав 2,2 секунди, однак середня тривалість очікування програми на обслуговування була близько 2 хвилин (тому що необхідна кількість процесорів перевищувала 64 і задача не могла бути вирішена в режимі оп-line). У даному підході було проведено (1:1) - відображення віртуальної моделі на реальну MIMD – систему.

Авторами створено програмне забезпечення, що реалізує запропоновані вище методи розподілу, проведені експерименти на ЕОМ



CRAY T3E стосовно до мережі рис. 1. Залежність часу моделювання від кількості використовуваних процесорів показано на рис.4. Чорним кольором представлені значення для рівномірного методу, сірим – для методу розподілу по антидереву. По осі ординат показаний час роботи програми, а по осі абсцис – кількість застосовуваних процесорів. Мінімум часу моделювання для методу рівномірного розподілу приходить на 24 процесора і складає 0,42 секунди. Для методу розподілу по гілках антидерева мінімум часу моделювання приходить на 22 процесора і складає 0,34 секунди.

Таким чином, розроблені методи розподілу можуть масштабуватись, мають більшу швидкість і вимагають застосування меншої кількості процесорів порівняно з (1:1) – відображенням (рис.3). Необхідно відзначити, що зі збільшенням кількості використовуваних процесорів понад 30 відбувається поступове збільшення часу моделювання, процесори починають «заважати» один одному, і час, затрачений на обмін даними між процесорами, збільшується.

## **6. Висновки**

Мережні динамічні об'єкти з зосередженими параметрами описуються матрично – векторними системами рівнянь, що у явній формі містять топологічний опис. Це дозволяє формалізувати побудову віртуальних MIMD – моделей об'єктів та їх девіртуалізацію.

Запропоновані методи розпаралелювання і девіртуалізації випробувані експериментально і є основою для побудови вирішувачів рівнянь у розподіленому паралельному моделюючому середовищі (РПМС). Результати модельних експериментів показують, що метод розподілу по гілках має переваги порівняно з іншими методами.

В процесі експериментальних досліджень вирішувачів рівнянь підтверджено надійне функціонування РПМС [4]. Це дозволяє розвивати подальші дослідження в таких напрямках: відпрацювання методики девіртуалізації моделей мережних об'єктів на різні цільові паралельні ресурси, що залучаються до складу РПМС; апробація MIMD-вирішувача на реальних складних вентиляційних мережах шахт Донбасу; паралельне моделювання систем автоматизованого керування шахтними вентиляційними мережами. Ці дослідження проводяться за планом наукового співробітництва ДонНТУ та Штутгартського університету.

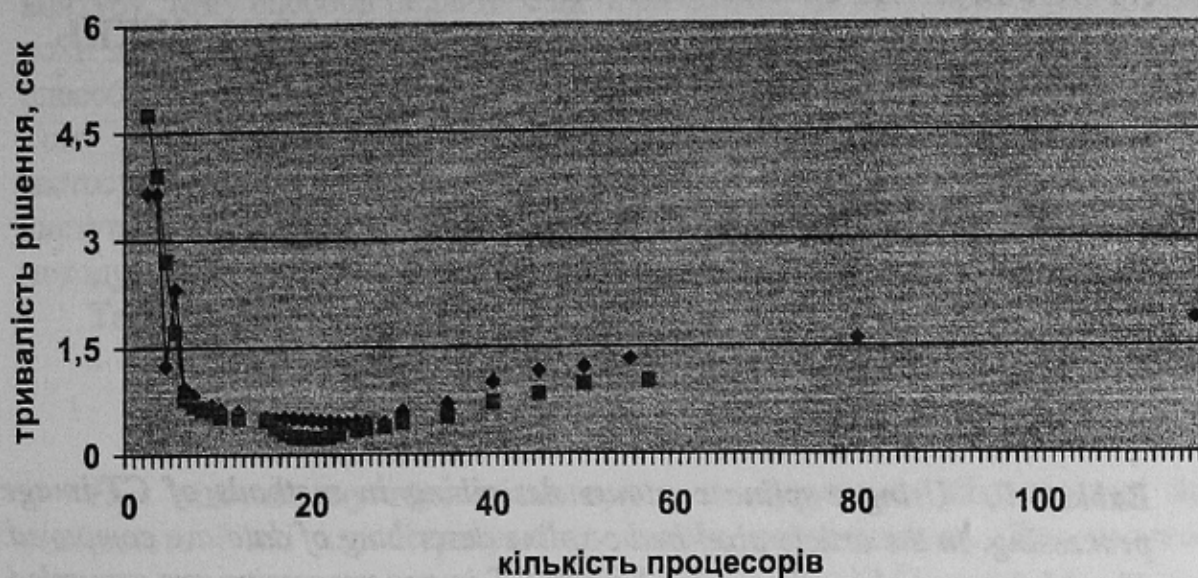


Рис. 4 – Порівняльний аналіз методів розпаралелювання

### Література

1. Абрамов Ф.А., Фельдман Л.П., Святный В.А. Моделирование динамических процессов рудничной аэрологии. Киев, Наукова думка, 1981г. 283с.
2. Святный В.А. Проблеми паралельного моделювання складних динамічних систем.- Наукові праці ДонДТУ, серія ІКОТ, вип. 6, Донецьк, 1999, С. 6-14.
3. Hanf G.: Modellierung und Simulation instantionären Grubenbewetterung auf verteilten Rechnerarchitekturen. VDI Verlag, Düsseldorf, 2002
4. Святный В.А., Солонін О.М., Надеєв Д.В., Степанов І., Ротермель К., Цайтц М. Розподілене паралельне моделююче середовище.- Наукові праці ДонДТУ. Серія "Проблеми моделювання та автоматизації проектування динамічних систем". Випуск 29:-Донецьк, ДонДТУ, 2001. – С. 229–234.
5. MPI: A Message-Passing Interface Standart Esprit Project P6643 (PPPE), 1994, 227р.

Дата надходження до редколегії: 23.12.2003 р.