

## ПРО ЗАСТОСУВАННЯ ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ В ЗАДАЧАХ МОДЕЛЮВАННЯ НА ОСНОВІ ІНДУКТИВНОГО ПІДХОДУ

*В.С. Степанко, С.М. Єфіменко*

Міжнародний науково-навчальний центр  
інформаційних технологій та систем НАН та МОН України  
03680, Київ, проспект Академіка Глушкова, 40,  
тел.: (044) 526 3028, astrid@irtc.org.ua, syefim@ukr.net

*О.П. Розенблат, А.І. Черняк*

Інститут програмних систем НАН України  
03187, Київ, проспект Академіка Глушкова, 40,  
тел.: (044) 526 6033, rozsa@isofts.kiev.ua, d2macster@gmail.com

Розглядається застосування паралельних обчислень з метою розширення можливостей моделювання за статистичними даними. Вивчаються питання організації паралельних обчислень в багатопроцесорних кластерних системах для розв'язування прикладних задач комбінаторного типу. Результати експериментів демонструють добрий ступінь розпаралелюваності та масштабованості обчислень з використанням комбінаторного алгоритму.

Parallel computation applications are considered with the purpose of extension of modeling possibilities based on statistical data. Issues concerning parallel computation organization in multiprocessor clusters are studied for solving applied problems of combinatorial type. The experimental results show a good degree of paralleling and scalability of computation using the combinatorial algorithm.

### Вступ

Для успішного розв'язання актуальних задач поточного управління соціально-економічними процесами в Україні необхідно здійснювати ґрунтовний кількісний аналіз цих процесів з метою їхнього прогнозування [1].

Наприклад, зрозуміло, що основні соціальні показники – заробітна платня, пенсії, зайнятість, рівень безробіття тощо – певною мірою пов'язані між собою. Зрозуміло також, що всі вони істотно залежать від процесів, які відбуваються в економічній, фінансовій, екологічній, демографічній та інших сферах. Тому задача прогнозування згаданих цільових показників вимагає, взагалі кажучи, побудови комплексних економіко-математичних моделей, що відображають найважливіші зв'язки з процесами, які відбуваються поза соціальною сферою, але є факторами істотного впливу на неї.

Проте усталені закономірності взаємодії зазначених процесів за умов України однозначно встановити важко, тому для розв'язання задач прогнозування слід застосовувати економетричні (економіко-статистичні) методи [2]. Основними етапами задачі при цьому є такі:

- збирання інформації (вибірки даних) про зміну в часі цільових показників та всіх факторів, передусім макроекономічних, які можуть впливати на ці показники;
- комплексний аналіз взаємозв'язку показників з факторами і виділення групи релевантних (найістотніших) факторів впливу на цільові показники;
- побудова на основі даних вибірки математичних моделей, що описують закономірності впливу релевантних факторів на цільові показники;
- застосування одержаних моделей для оцінювання майбутніх значень цільових показників за тих чи інших варіантів змінювання групи істотних факторів (нормативне прогнозування).

Для виконання цих завдань можна застосовувати поширені засоби математичної статистики – методи факторного, дисперсійного та регресійного аналізів, проте вони потребують значної апріорної інформації. Тому рекомендується застосовувати індуктивний підхід до розв'язання задач побудови прогнозних моделей, який за рахунок автоматизації процедур інтенсивного комп'ютерного пошуку адекватних моделей на базі вибірки (сховища) даних дозволяє значно прискорити процес моделювання. Методологія цього підходу ґрунтується на максимальному “видобуванні” всієї необхідної інформації з вибірки даних і орієнтована на включення в модель лише найістотніших у конкретних умовах факторів, а не всіх, які можуть впливати на цільовий показник. При цьому, на відміну від поширених підходів, відпадає необхідність задавати структуру моделі.

Такий підхід реалізовано в методі групового урахування аргументів (МГУА [3], англійська назва – Group Method of Data Handling (GMDH)), який є методом виявлення прихованих закономірностей з автоматичним вибором оптимальної структури і параметрів моделі на основі інформації вибірки статистичних даних. Цей метод призначений для побудови лінійних, нелінійних, різницевих та інших математичних моделей складних процесів за короткими вибірками даних. В його основу покладено принципи зовнішнього доповнення, автоматичної генерації та послідовної селекції ускладнюваних структур моделей. Проте перебірний характер

цього методу часто не дозволяє розв'язувати задачі моделювання з великим числом аргументів (факторів). Тому в цій доповіді розглядаються питання застосування кластерних технологій з метою розширення можливостей моделювання за статистичними даними.

## 1. Постановка задачі моделювання за даними спостережень

У найпростішому вигляді (у випадку лінійної залежності) задача побудови залежності між певними статистичними показниками полягає в побудові функції

$$y = f(\theta, x) = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_m x_m, \quad (1)$$

де  $y$  – цільовий показник,  $x_i, i = \overline{1, m}$  –  $m$  факторів, які впливають на цей цільовий показник.

Залежність (1) можна подати у вигляді

$$y = X\theta, \quad (2)$$

де  $\theta$  – невідомий вектор коефіцієнтів моделі, тобто попередньо неможливо зазначити, з якими з цих факторів найбільше зв'язаний цільовий показник, або, іншими словами, яка група факторів впливає найбільше на цільовий показник.

Значимо, що в (1) під факторами  $x_i, i = \overline{1, m}$  можна розуміти також будь-які функції від певних вимірюваних незалежних показників – наприклад, квадрати, змішані добутки, інші нелінійні перетворення, запізнювані та лагові значення тощо, тобто (1) можна розглядати як загальну форму функції, лінійної за параметрами.

Задача полягає у тому, щоб знайти невідомий вектор коефіцієнтів  $\theta$ , тобто ту підмножину факторів, яка найкраще описує цільовий показник. Така підмножина може бути різної величини – від 1 до  $m$ . Але які саме фактори входять до цієї підмножини, наперед невідомо, адже, наприклад, коефіцієнти парної кореляції між цільовим показником та окремими факторами не дають підстав для виявлення характеру (структури) залежності, тому що, як правило, істотний вплив має багатовимірне взаємодія між факторами. Так, економічні показники дуже тісно пов'язані між собою, тому внутрішні взаємозв'язки між факторами теж впливають на оптимальну структуру моделі.

Таким чином, отримуємо задачу побудови найкращої моделі, тобто оптимальної в розумінні мінімуму певного заданого критерію якості моделей. Хоча це й важливо з прикладної точки зору, але не має значення в контексті застосування паралельних обчислень.

Виходячи з вигляду загальної моделі, можна сконструювати багато варіантів моделей, які мають різну складність, тобто різний склад і кількість факторів. Найбільш очевидним варіантом формування цієї множини всіх можливих варіантів моделей є така послідовність дій. Побудувати  $C_1^m$  залежностей цільового показника у

від факторів, коли модель містить тільки один аргумент, потім  $C_2^m$  моделей, що містять 2 аргументи і т.д. до

$C_m^m = 1$  моделі, що включає всі аргументи. Якщо знайти суму  $\sum_{i=1}^m C_i^m$ , то отримаємо загальну кількість

можливих варіантів моделей, яка буде показниковою функцією від кількості аргументів:  $2^m - 1$ . Таким чином, для повного перебору слід побудувати загалом  $2^m - 1$  моделей. Це означає, що спочатку необхідно сформулювати структуру кожної часткової моделі, потім оцінити її параметри, а далі обчислити відповідне їй значення критерію якості. Зокрема, у найпростішому випадку, при використанні традиційного для регресійного аналізу методу найменших квадратів, розв'язок задачі оцінювання має вигляд

$$\hat{\theta} = (X^T X)^{-1} X^T y. \quad (3)$$

Це задача використання даних статистики. Ми формуємо множину моделей і для кожної з них оцінюємо параметри на основі статистики. Цей процес називається генерацією варіантів. При цьому моделі генеруються не всі одразу, а поступово, з відбором у поточному режимі найкращої за значенням заданого критерію, який треба мінімізувати на всій множині  $2^m - 1$  моделей.

Таким чином, розв'язання задачі відбувається за допомогою повного перебору варіантів. При цьому очевидною перевагою повного перебору є те, що не виключається жоден з можливих варіантів. Недоліком може бути досить значний час виконання. Вимогою до отримання розв'язку є прийнятний час виконання, який не повинен перевищувати кількох хвилин. Тому до недавнього часу можна було використовувати невелику кількість аргументів – повний перебір був доцільним при  $m < 20$  факторів. Це пояснюється великими обчислювальними затратами, оскільки при додаванні одного фактора час виконання зростає удвічі. При проведенні експериментів на сучасних ЕОМ було встановлено, що повний перебір доцільний при  $20 \div 25$  факторах (залежить від потужності процесора ЕОМ). При застосуванні спеціальних схем, які дозволяють значно покращити швидкодію алгоритмів, різниці у швидкодії в порівнянні зі звичайними методами складає десятки разів, що є свідченням ефективності таких модифікацій. Та все ж при використанні таких ефективних схем повний перебір доцільно використовувати при кількості аргументів орієнтовно  $< 30$ . При 50 змінних ніякий повний перебір на однопроцесорній ЕОМ неможливий.

## 2. Особливості застосування паралельних обчислень

Саме з точки зору розширення можливостей повного перебору можна в цій задачі застосувати багатопроцесорний комплекс. Виходячи з кількості моделей  $2^m$ , що формуються при повному переборі, очевидно є ідея розпаралелювання.

**2.1. Найпростіший випадок кількості процесорів.** У найпростішому випадку, коли число процесорів дорівнює  $p = 2k$ , де  $k$  – деяке ціле число (тобто такий багатопроцесорний комплекс складається з 2, 4, 8, 16 і т.д. однопроцесорних машин), то легко розділити загальне число  $2^m$  на однакову кількість моделей, які будуть оброблятися кожним незалежним процесором. Так, якщо маємо 2 процесори, то на кожний з них припадає  $2^{m-1}$  варіантів моделей. При кількості процесорів, рівній  $2k$ , кожен окремий процесор будуватиме  $2^{m-k}$  моделей.

Очевидно, що за загальної кількості процесорів  $2k$  коефіцієнт  $k$  буде показувати, скільки „додаткових” аргументів зможе обрахувати багатопроцесорний комплекс у порівнянні з однопроцесорною машиною за умови, що потужність усіх задіяних процесорів рівна.

**2.2. Випадок довільної кількості процесорів.** Складнішим виглядає загальніший випадок, коли на кількість процесорів  $p$  не накладаються ніякі умови, тобто вона може бути довільною. У такому разі можна запропонувати два варіанти розбиття загальної кількості усіх можливих моделей, що перебираються:

Десяткове число, яке відповідає кількості кроків  $2^m$ , ділиться на кількість процесорів  $p$  націло, а остача припадає на останній процесор. При цьому на нього припадатиме у загальному випадку дещо менше обчислювальне навантаження, яке відрізнятиметься від решти процесорів не більше, ніж на  $p$  додаткових моделей. Однак за умови, що  $2^m \gg p$ , зменшення часу моделювання можна знехтувати.

Нерівномірність у обчислювальному навантаженні розподіляється між декількома процесорами, і тоді менша кількість (але при цьому не більше ніж на одну додаткову модель) моделей припадатиме не на один останній процесор, а на декілька.

**2.3. Випадок різної потужності процесорів багатопроцесорного комплексу.** Найскладнішим видається випадок, коли багатопроцесорний комплекс складається з  $p$  кластерів різної потужності. Нехай  $t_i$ ,  $i = \overline{1, p}$  – потужність  $i$ -го кластера, яку в разі процесорів одного типу (наприклад, Intel Pentium), можна представити числом, пропорційним їх частоті. Тоді можна запропонувати обчислити сумарну потужність усіх

задіяних кластерів  $T = \sum_{i=1}^p t_i$ , після чого визначити „відносний коефіцієнт потужності”  $i$ -го кластера як відношення  $t_i/T$ . Обчислювальне навантаження на кожен кластер слід розподіляти пропорційно до обчисленого таким способом його коефіцієнта потужності. Зрозуміло, що перед проведенням експериментів необхідно мати детальну інформацію щодо властивостей кожного кластера багатопроцесорного комплексу.

Комплекс відповідних програм складається з декількох блоків: отримання початкової інформації, обробка інформації для формування необхідних масивів, генерація структур, оцінювання параметрів кожної зі згенерованих структур, обчислення критерію якості. Поточний відбір моделі, якій відповідає оптимальне значення заданого критерію, здійснюється таким чином, що не запам'ятовуються всі моделі, а відслідковується одна або декілька найкращих, які порівнюються з наступними згенерованими. Так, при відборі однієї найкращої моделі, якщо наступна згенерована модель є кращою за попередні, то вона запам'ятовується, якщо ні – то відкидається.

Якщо загальну кількість варіантів  $2^m$  поділити на рівні частини, то відповідна кількість генераторів буде утворювати структури, не змішувани між собою, і при цьому буде гарантовано, що в цій сукупності паралельні генератори структур даватимуть повний набір різних моделей.

Можливо, необхідною є певна головна програма (монітор), яка для кожної групи моделей буде відслідковувати кращу (або кращі) з них та у поточному режимі запам'ятовуватиме кращу (або кращі) модель серед усіх згенерованих усіма процесорами.

## 3. Кластерні системи

На сьогоднішній день не складає значних труднощів створити невелику кластерну систему, об'єднавши декілька комп'ютерів лабораторії або навчального класу.

Кластерні технології дозволяють для досягнення необхідної продуктивності поєднувати в єдині обчислювальні системи комп'ютери зовсім різного типу, починаючи від персональних комп'ютерів і закінчуючи потужними суперкомп'ютерами. Кластерні технології широко використовуються як засіб створення систем суперкомп'ютерного класу зі складових частин масового виробництва, що значно зменшує вартість обчислювальної системи. Зокрема, одним з перших був реалізований проект COSOA [4], у якому на базі 25 двохпроцесорних персональних комп'ютерів загальною вартістю порядку \$100000 була створена система із продуктивністю, еквівалентній 48-процесорному Cray T3D вартістю кілька мільйонів доларів США [5].

Звичайно, про повну еквівалентність цих систем свідчити не доводиться. Комунікаційне середовище можна досить повно охарактеризувати двома параметрами: *латентністю* – часом затримки при посилці повідомлення, і *пропускною здатністю* – швидкістю передачі інформації. Отож, для комп'ютера Cray T3D ці параметри становлять відповідно 1 мкс і 480 Мб/сек, а для кластера, в якому як комунікаційне середовище

використано мережу Fast Ethernet, 100 мкс і 10 Мб/сек. Це почасти пояснює дуже високу вартість суперкомп'ютерів. При таких параметрах, як у розглянутого кластера, знайдеться не так багато завдань, які можна ефективно розв'язувати на досить великій кількості процесорів.

Якщо свідчити коротко, то *кластер* – це пов'язаний набір повноцінних комп'ютерів, використовуваних у якості єдиного обчислювального ресурсу. Переваги кластерної системи перед набором незалежних комп'ютерів очевидні. По-перше, розроблено безліч диспетчерських систем пакетної обробки завдань, що дозволяють послати завдання на обробку кластеру в цілому, а не якомусь окремому комп'ютеру. Ці диспетчерські системи автоматично розподіляють завдання по вільних обчислювальних вузлах або буферизують їх за відсутності таких, що дозволяє забезпечити більше рівномірне й ефективне завантаження комп'ютерів. По-друге, з'являється можливість спільного використання обчислювальних ресурсів декількох комп'ютерів для розв'язання одного завдання.

Для створення кластерів звичайно використовуються або прості однопроцесорні персональні комп'ютери, або двох-, чотири- процесорні SMP-сервери. При цьому не накладається ніяких обмежень на склад та архітектуру вузлів. Кожний з вузлів може функціонувати під керуванням своєї власної операційної системи. Найчастіше використовуються стандартні операційні системи: Linux, FreeBSD, Solaris, Windows NT. У тих випадках, коли вузли кластера неоднорідні, свідчать про *гетерогенні* кластери.

При створенні кластерів можна виділити два підходи. Перший підхід застосовується щодо невеликих кластерних систем. У кластер поєднуються повнофункціональні комп'ютери, які продовжують працювати і як самостійні одиниці, наприклад, комп'ютери навчального класу або робочі станції лабораторії. Другий підхід застосовується в тих випадках, коли цілеспрямовано створюється потужний обчислювальний ресурс. Тоді системні блоки комп'ютерів компактно розміщуються в спеціальних стійках, а для керування системою та запуску завдань виділяється один або декілька повнофункціональних комп'ютерів, які називаються хост-комп'ютерами. У цьому випадку немає необхідності забезпечувати комп'ютери обчислювальних вузлів графічними картами, моніторами, дисковими накопичувачами й іншим периферійним обладнанням, що значно здешевлює вартість системи.

Розроблено безліч технологій поєднання комп'ютерів у кластер. Найбільш широко нині використовується технологія Fast Ethernet. Це обумовлено простотою її використання й низькою вартістю комунікаційного обладнання. Однак за це доводиться свідомо розплачуватися недостатньою швидкістю обмінів. Справді, це обладнання забезпечує максимальну швидкість обміну між вузлами 10 Мб/сек, тоді як швидкість обміну з оперативною пам'яттю становить 250 Мб/сек і вище. Розроблювачі пакета підпрограм ScaLAPACK, призначеного для розв'язання задач лінійної алгебри на багатопроцесорних системах, у яких велика частка комунікаційних операцій, формулюють у такий спосіб вимогу до багатопроцесорної системи: "Швидкість міжпроцесорних обмінів між двома вузлами, вимірювана в Мб/сек, повинна бути не менш 1/10 пікової продуктивності обчислювального вузла, вимірюваної в Mflops"[6]. Таким чином, якщо як обчислювальні вузли використати комп'ютери класу Pentium III 500 МГц (пікова продуктивність 500 Mflops), то апаратура Fast Ethernet забезпечує тільки 1/5 від необхідної швидкості. Частково це положення може поправити перехід на технології Gigabit Ethernet.

Найбільш доступний спосіб створення обчислювальних ресурсів суперкомп'ютерного класу надають кластерні технології. У найпростішому випадку за допомогою мережних комунікаційних пристроїв поєднуються однотипні комп'ютери, і в їхнє програмне забезпечення додається комунікаційна бібліотека типу MPI. Це дозволяє використати набір комп'ютерів як єдиний обчислювальний ресурс для запуску паралельних програм. Однак напевно таку систему можна назвати повноцінним кластером. Виявляється, що для ефективної експлуатації такої системи конче необхідна певна диспетчерська система, яка б розподіляла завдання користувачів за обчислювальними вузлами і блокувала б запуск інших завдань на зайнятих вузлах [7].

Коллектив Інституту кібернетики ім. В.М. Глушкова НАН України завершив роботу над створенням суперкомп'ютера для реалізації інформаційних технологій (СКИТ) – ефективного розв'язання надскладних математичних задач, що виникають у різних сферах людської діяльності.

Програмні й апаратні засоби дозволяють вирішувати одну задачу з використанням усього обчислювального ресурсу, а також розділяти поле для рішення (сукупність усіх вузлів кластера) на частини необхідного розміру й надавати їх декільком користувачам.

Обчислювальні засоби кластера (вузли й керуюча ЕОМ) являють собою масив обчислювальних модулів, пов'язаних між собою трьома локальними обчислювальними мережами – високошвидкісними мережами SCI, Gigabit Ethernet, Fast Ethernet.

Мережа SCI призначена для високошвидкісного обміну між вузлами в ході обчислень. При обміні даними між двома вузлами по мережі SCI з використанням протоколів MPI може бути досягнута пропускна здатність 345 Мбайт/с.

Мережа Gigabit Ethernet призначена для з'єднання всіх вузлів кластера (розв'язувального поля) з керуючою ЕОМ і файлом-сервером. Мережа Fast Ethernet призначена для початкового завантаження програм і даних у вузол та для передачі службової інформації про хід обчислювального процесу й стан його апаратури.

16-вузловий кластер заснований на 32-бітній архітектурі Intel IA32 (x86) (СКИТ-1), 32-вузловий кластер – на 64-бітній архітектурі Intel (ia64) (СКИТ-2) [8].

#### 4. Інтерфейс паралельного програмування MPI

Нині однією з найпоширеніших технологій програмування для паралельних комп'ютерів з розподіленою пам'яттю є MPI. Основним способом взаємодії паралельних процесів у таких системах є взаємна передача повідомлень, що й відбито в назві цієї технології – Message Passing Interface (інтерфейс передачі повідомлень).

MPI-програма являє собою набір незалежних процесів, кожний з яких виконує свою власну програму (не обов'язково ту саму), написану мовою C або FORTRAN. Процеси MPI-програми взаємодіють один з одним за допомогою виклику комунікаційних процедур. Як правило, кожен процес виконується у своєму власному адресному просторі, однак допускається й режим поділу пам'яті. MPI не специфікує модель виконання процесу – він може бути як послідовним, так і багатопотоковим.

Отже, якщо сформулювати коротко, MPI – це бібліотека функцій, що забезпечує взаємодію паралельних процесів за допомогою механізму передачі повідомлень. Це досить об'ємна й складна бібліотека, що складається приблизно з 130 функцій, до числа яких входять:

- функції ініціалізації й закриття MPI процесів;
- функції, що реалізують комунікаційні операції типу точка-точка;
- функції, що реалізують колективні операції;
- функції для роботи із групами процесів і комунікаторами;
- функції для роботи зі структурами даних;
- функції формування топології процесів.

Системи, де кожен процес виконує ту саму програму (параметризовану щодо ідентифікатора чи процесу, або процесора), але працює з різними даними, називають моделлю програмування SPMD (single program multiple data - одна програма, багато даних). SPMD-модель прийнятна й досить зручна для широкого діапазону застосувань паралельного програмування, але вона утруднює розробку деяких типів паралельних алгоритмів.

#### 5. Експерименти з паралельних обчислень

Загальний алгоритм розв'язання задачі застосування кластерних технологій для моделювання на основі індуктивного підходу показано на рис. 1.

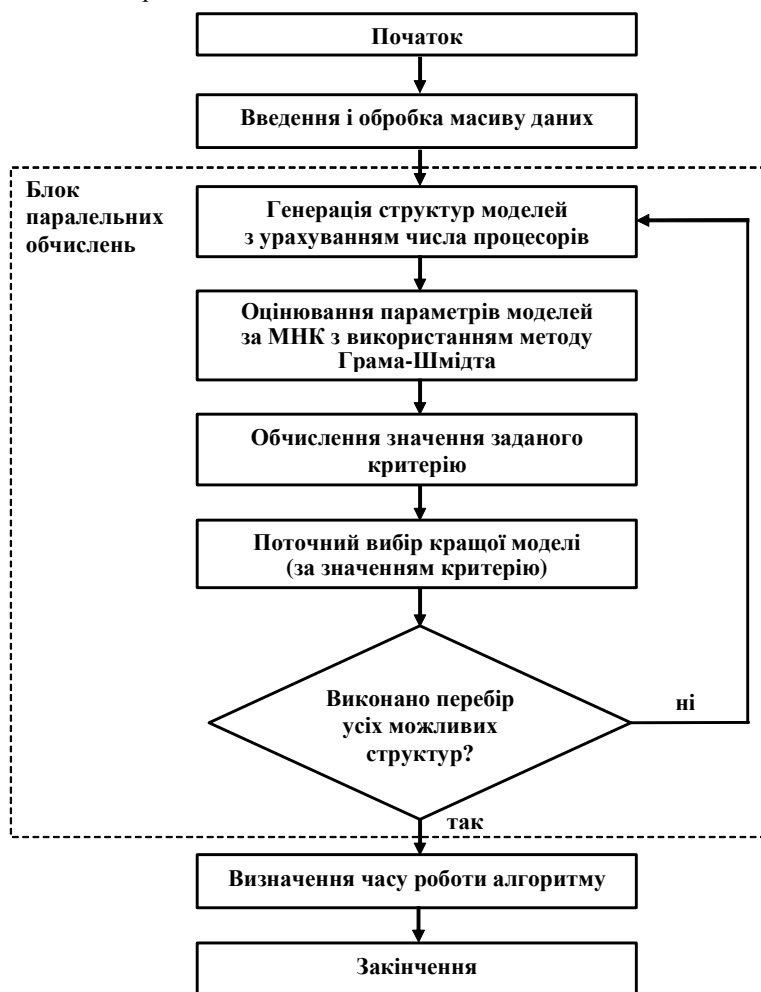


Рис. 1. Блок-схема комбінаторного алгоритму

Було виконано тестові експерименти з розпаралелювання задачі. Нижче на рис. 2. показано діаграму часу виконання програми (вісь Z) на різних кількостях процесорів (від 1 до 8, вісь X) за різного числа факторів (числа стовпців матриці, вісь Y). Результати показують добрий ступінь розпаралелюваності та масштабованості обчислень з використанням комбінаторного алгоритму.

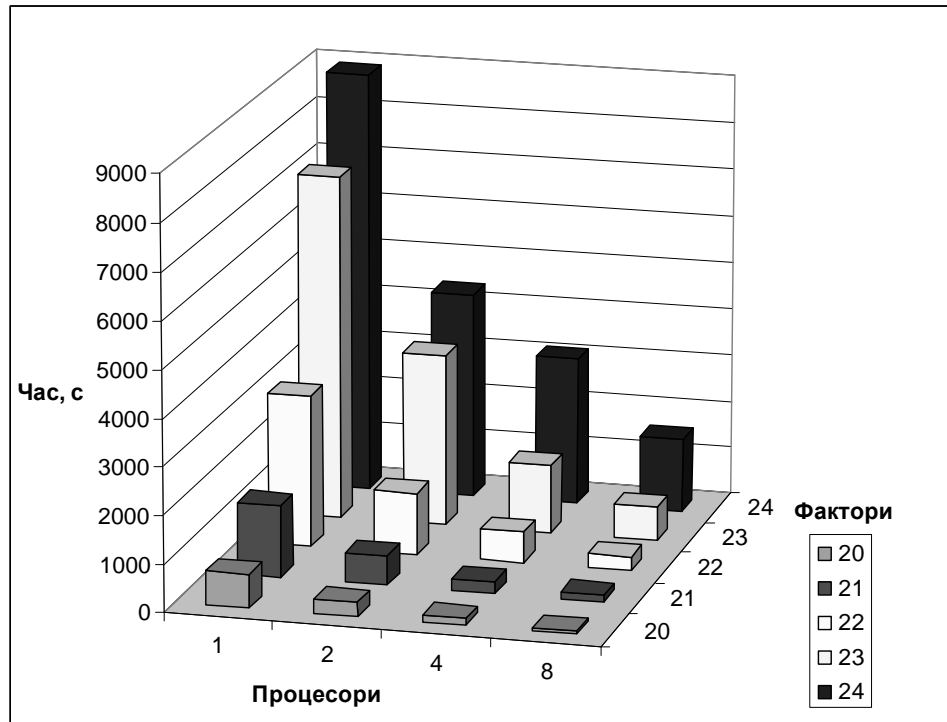


Рис. 2. Діаграма часу виконання програми

На рис. 3 показано залежність величини мультипроцесорного прискорення від кількості використовуваних процесорів [9], що оцінюється за формулою  $s = \frac{T_1}{T_n}$ , де  $T_1$  – час розв’язання задачі на одному процесорі, а  $T_n$  – час розв’язання тієї ж задачі на  $n$  процесорах.

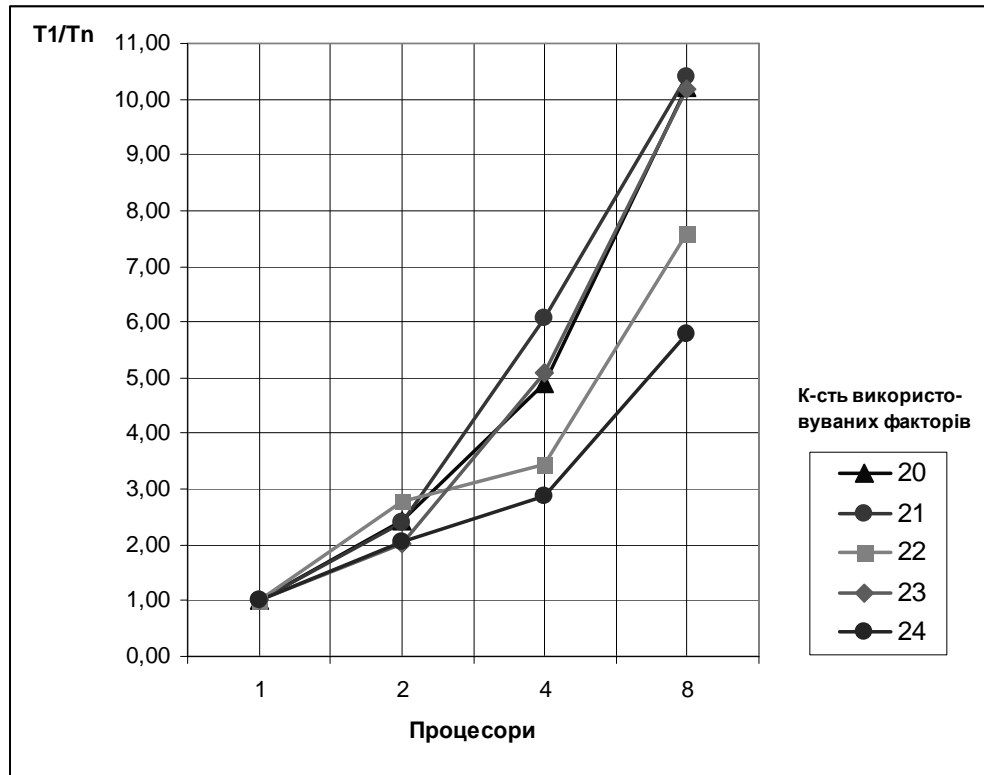


Рис. 3. Залежність мультипроцесорного прискорення від кількості використовуваних процесорів

## **Висновки**

Розглянуто застосування паралельних обчислень з метою розширення можливостей моделювання за статистичними даними. Досліджено також питання організації паралельних обчислень у кластерних системах на прикладі розв'язування прикладних задач комбінаторного типу. Наведено результати експериментів, які демонструють добрий ступінь розпаралелюваності та масштабованості обчислень з використанням комбінаторного алгоритму.

1. *Гесць В., Скрипниченко М., Соколик М., Шумська С.* Секторальні макромоделі прогнозування економіки України // Економіст. – 1998. – № 5. – С. 58–67.
2. *Лукінов І., Бакаєв О., Бондаренко Г.* Системи макроеконометричних моделей прогнозування економіки України // Економіст. – 1998. – № 5. – С. 38–46.
3. *Ивахненко А.Г., Степанко В.С.* Помехоустойчивость моделирования. – Киев: Наук. думка, 1985. – 216 с.
4. *The Cost Effective Computing Array (COCOА).* – <http://cocoa.aero.psu.com>
5. *The Cray3D.* – <http://www.cray-cyber.org/systems/t3d.php>
6. *ScaLAPACK Users' Guide.* 1997. – [http://www.netlib.org/scalapack/scalapack\\_home.html](http://www.netlib.org/scalapack/scalapack_home.html) [http://rsusu1.rnd.runnet.ru/parallel/scalapack/scalapack\\_home.html](http://rsusu1.rnd.runnet.ru/parallel/scalapack/scalapack_home.html)
7. *Букатов А.А., Пацюк В.Н., Жегуло А.И.* Программирование многопроцессорных вычислительных систем. Ростов-на-Дону. Из-во ООО «ЦВВР», 2003. – 208 с.
8. *Суперкомп'ютери* Інституту Кібернетики ім. В.М. Глушкова НАН України: СКІТ-1 СКІТ-2. – <http://cluster.icyb.kiev.ua/>
9. *Дорошенко А.Е.* Математические модели и методы организации высокопроизводительных параллельных вычислений. Алгебродинамический подход.– Киев: Наук. думка, 2000.