Issue 6

https://doi.org/10.23939/sisn2019.02.066

УДК 510.67(07)

АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ФУНКЦІОНУВАННЯ СКАЛЯРНИХ БАГАТОПРОЦЕСОРНИХ СИСТЕМ

І. Р. Опірський[1], С. В. Зибін[2], В. О. Хорошко[3]

[1] Національний університет "Львівська політехніка", кафедра захисту інформації
[1] iopirsky@gmail.com, ORCID: 0000-0002-8461-8996
[2, 3] Національний авіаційний університет, кафедра безпеки інформаційних технологій
[2] professor_va@ukr.net, ORCID: 0000-0001-6213-7086
[3] duikt@ua.fm, ORCID: 0000-0002-2670-2823

Досліджено математичні моделі функціонування складних систем. Встановлено відповідність результатів дослідження якісним уявленням про поведінку системи, насичення системи у разі збільшення кількості користувачів.

Побудовано модель однорідної складної системи за наявності пріоритетного (потоку) великих задач. Досліджено функціонування однорідної складної системи під час опрацювання потоків складних задач за припущення, що у систему із *N* елементарних персональних електронно-обчислювальних машин з додатковою нескінченною зовнішньою пам'яттю надходить пуассонівський потік програм (Р-програми) із певною частотою.

Запропоновано новий підхід до дослідження функціонування складних систем. Передбачено, що система має різнотипні ресурси та заявки, що потребують різних комбінацій цих ресурсів. У роботі запропонована доволі детальна модель системи, яку названо багаторесурсною системою масивного обслуговування.

Запропоновано методику визначення межі потужності багаторесурсної системи масивного обслуговування. Ця методика дає змогу визначити межі потужності для системи із фіксованими характеристиками вхідного потоку. Запропоновано модель, що дає можливість встановлювати мінімальну необхідну кількість джерел запитів за заданої кількості процесорних зон робочого поля. Коли перерозподіл ресурсів здійснюється тактовно, то описану модель можна застосовувати для вибору мультипрограмування як для мультипроцесорних (розподіл процесорів між задачами), так і для мультипрограмних однопроцесорних обчислювальних систем (розподіл оперативної пам'яті між задачами). Запропоновано методику підрахунку кількості варіантів розміщень задач в обчислювальній системі в межах цієї структури чи множини структур, а також методику визначення вірогідних характеристик однорідних складних систем у режимі пам'ятного опрацювання складних задач за різних дисциплін розподілу. Вона придатна для параметрів, характерних для проєктування сучасних однорідних складних систем, які повинні містити сотні й тисячі мініпроцесорів і обслуговувати сотні користувачів.

Проаналізовано математичні моделі щодо функціонування складних систем, що дозволяє здійснювати вибір тієї чи іншої системи для конкретної задачі. Отримані результати узгоджуються з уявленнями про поведінку систем та насичення системи у разі збільшення кількості користувачів. Крім того, виявлено вірогідність розв'язання усіх задач системою, що дає змогу прогнозувати продуктивність однорідних складних систем і використовувати для вибору параметрів проєктованих систем.

Ключові слова: багатопроцесорна система, однорідна складна система, багаторесурсна система масового обслуговування, процес розподілу ресурсів, опрацювання потоків, вірогідність, програми рангу, межа потужності.

Вступ

Розвиток систем обчислювальної техніки зумовив тенденцію до застосування паралельних багатопроцесорних систем (БПС), оскільки межі підвищення швидкодії електронно-обчислювальних машин (ЕОМ), що входять до таких систем, вже практично досягнуті. Відповідно, надалі підвищувати їх швидкодію можливо тільки за рахунок паралельного опрацювання задач. Як показує сучасний досвід використання однорідних складних систем (ОСС), до складу яких входять БП обчислювальні системи (ОС) [1], їх перспективу можна розширити тільки підвищенням ефективності організації їх функціонування за рахунок використання нових сучасних методів та засобів.

Щоб виконати оцінку системи і розробити її математичні моделі, визначимо характерні для неї особливості:

1) об'ємність і складність моделей системи, що вимагає використання для її аналізу сучасного математичного апарату, моделювання і методів щодо її побудови;
2) взаємопов'язаність внутрішніх підсистем і елементів;
3) багатовимірність, зумовлена великою кількістю наявних зв'язків і взаємодій у підсистемах;
4) зв'язок із навколишнім середовищем;
5) робота під час впливу незалежних та залежних довільних випадкових факторів;
6) унеможливлення визначення інформації про характеристики системи, а також її властивостей за рахунок вивчення тільки характеристик її елементів;
7) велика кількість критеріїв щодо оцінки якості функціонування СС і її компонентів;
8) висока складність функцій, які виконує система;

Задачі знаходження оптимальних ресурсів (за деяким вибраним критерієм якості) належать до екстремальних задач із комбінаторики [2]. Їх забезпечення є метою функціонування ОСС, щодо якої вибирають відповідний критерій якості. Отже, СС – це множина підсистем та елементів різноманітної природи, які взаємодіють та взаємопов'язані, забезпечують розв'язання складної задачі й описують складну математичну модель.

Постановка проблеми

Складні ОС із програмно конфігурованою структурою зайняли належне місце у БП розподілених системах, функціонування яких основане на використанні децентралізованих алгоритмів керування. Під час моделювання складних БПС і для вирішення поставлених завдань можна скористатися моделлю багаторесурсної системи масового обслуговування (БСМО). У цьому випадку процесори уподібнюються до каналів обслуговування, а їх задачі – до заявок на виконання робіт. Для того, щоб оцінити їх функціональний стан, необхідно застосовувати методи моделювання та методи діагностування. Поява СС, до яких входять комплекси (системи), що адаптуються до задач, тобто перелаштовують свою конфігурацію залежно від дій та запитів користувачів або поставлених задач, зумовила потребу в розробленні нових скалярних математичних моделей систем зі сконфігурованою програмно структурою. Крім цього, ці системи дають змогу здійснювати контроль функціональних характеристик як самих систем, так і сусідніх. У цьому випадку стан системи контролюється в умовах множини несправностей завдяки порівнянню результатів перехресної перевірки.

Тому створення математичних моделей складних БПС з програмно конфігурованою структурою дасть змогу розв'язати багато задач, які виникають сьогодні.

Аналіз останніх досліджень та публікацій

Запропоновано класифікації паралельних систем, призначені для різних цілей. Для розроблення паралельних алгоритмів важливо знати типи оперативної пам'яті (ОП), оскільки вони відображають способи взаємодії між компонентами паралельних програм. Ці процесори розділені на такі класи залежно від організації ОП:

1) із розподіленою пам'яттю – системи, процесорам в яких належить власна ОП, до якої не мають доступу інші процесори;

2) зі спільною пам'яттю – мультипроцесори, яким належить одна віртуальна пам'ять. В таких системах усі процесори отримують рівноцінний доступ до необхідних даних та/або команд, що опрацьовуються у пам'яті.

Класифікація, яку запропонував М. Флінн [5], основана на "понятті потоку", а саме тих команд чи/або даних, що опрацьовує процесор. Відповідно можна виокремити такі класи архітектур на основі кількості потоків [6–9]:

1) однопотокові SISD;
2) комбіновані SIMD;
3) комбіновані MISD;
4) множинні на команди і дані MIMD.

Сучасні "серйозні" персональні комп'ютери, смартфони, обчислювальна техніка, ноутбуки тощо реалізовані в класі MIMD-архітектури, тому дослідження з цієї тематики актуальні й багато науковців працюють, щоб підвищити ефективність функціонування цієї технології. Однак, незважаючи на значну кількість досліджень [6–11], є багато невирішених завдань, серед яких питання створення та аналізу математичних моделей складних БПС із програмно конфігурованою структурою.

Основна частина

Побудуємо модель ОСС за наявності пріоритетного потоку великих та об'ємних задач. Передбачаємо, що на ОСС з l ЕОМ надходить пуассонівський потік складних задач із параметром λ. Вірогідність надходження задачі з рангом k позначимо як b_k, причому $\sum_{k>\frac{l}{2}} b_k = 1.$ Задачу вважатимемо складною, коли її ранг перевищує $\frac{l}{2}$, де l – кількість ЕОМ у системі.

Обслуговування задач здійснюється у послідовності надходження їх у систему. Закон обслуговування є довільною задачею і реалізується для відповідного рангу поетапно. Система перебуває у деякому стані i, якщо i задач у черзі в процесорі. Вірогідність $G_{ij}(x)$ переходу системи зі стану i в j за час x пропонуємо визначати за такими співвідношеннями:

$$G_{ij}(x) = \sum_{v>\frac{l}{2}} b_v \int_0^x [1 - e^{\lambda(x-y)}] e^{-\lambda y} \frac{(\lambda y)^j}{j!} dH_v(y), \quad (1)$$

$$G_{ij}(x) = \sum_{v>\frac{l}{2}} b_v \int_0^x e^{-\lambda y} \frac{(\lambda y)^j}{j!} dH_v(y), \quad (2)$$

$$G_{ij}(x) = \sum_{v>\frac{l}{2}} b_v \int_0^x [1 - e^{\lambda(x-y)}] e^{-\lambda y} \frac{(\lambda y)^{j-i+1}}{(j-i+1)!} dH_v(y), \quad (3)$$

де $H_v(y)(v > \frac{l}{2})$ – функція розподілу часу для розв'язання задач v-го рангу.

За допомогою генератрис отримаємо відповідні умови, за яких період завантаженості системи має скінченну тривалість. Періодом завантаженості вважатимемо різницю між t_k (момент часу, коли система із завантаженого стану переходить у вільний) і найтривалішим, таким, що не перевершує його, t'_k (момент часу переходу із вільного стану в завантажений).

Дослідимо функціонування ОСС під час опрацювання потоків складних задач, припускаючи, що в систему з N елементарних ЕОМ (ЕМ) з додатковою нескінченною зовнішньою пам'яттю надходить пуассонівський потік програм (P-програми) із інтенсивністю a. Вірогідність надходження P-програми рангу n є a_n ($\sum_{n=1}^{R} a_n = 1, a_n > 0, R$ - максимальний ранг програм, які надходять у систему). Час обслуговування P-програм можна визначити за експоненціальним законом із параметром β. Приймемо, що в ОСС можуть одночасно існувати підсистеми всіх рангів від 1 до R, тобто $N \geq \frac{R(R+1)}{2}$. Введемо в ОСС множину з $k \in E (E = \{0,1,...,N\})$ M_n і розділимо її на різні підсистеми. Кількість підсистем рангу n, що створені у момент часу t зазначеним розподілом, позначимо як $l_k(t)$, а $l'_s(t)$ – кількість P-програм рангу n, що очікують в черзі з довжиною s, у момент часу t. Введемо для поставленої задачі деякий випадковий процес $W(t) = \{l_{k,n}(t), l'_{s,n} \frac{t}{k} \in E, n = \overline{1,k}, s = 0, \infty\}$. Згідно із прийнятими припущеннями він вважатиметься марковським. Нехай $P_k^{l_1,...,l_k,l'_1,...,l'_k}(t)$ – вірогідність того, що в момент часу t $k_1 \left[\sum_{n=1}^{R} nl_{k_1}, n(t)\right]$ ЕМ зайнято обслуговуванням P-програм. Крім того, для розв'язання нашої задачі необхідно ввести і обчислити: P_{kN} – вірогідність того, що в ОСС є рівно k гілок; $N_{cp} = \sum_{k=1}^{N} kP_k + N\sum_{s=1}^{\infty} P_{N+s}$ – середня кількість зайнятих шин процесора; $k_з(N) = \frac{N_{cp}}{N}$ – коефіцієнт ОСС, де $p_k = \sum P_k^{l_1,...,l_k,l'_1,...,l'_k} = \sum \lim_{t \to \infty} P^{l_1,...,l_{R'(t)},l'_1,...,l'_k}$ (підсумовування здійснюється за всіма можливими ЕМ на підсистему) за припущення, що виконується така умова нормування – $\sum_{k=0}^{\infty} k = 1$.

Нехай $N_{k,R}$ – кількість методів, за допомогою яких можна виконати розбиття k ЕМ P-програмами, максимальний ранг яких дорівнює R. Тоді, підсумувавши $N_{k,R}$ разів за $P_k^{l_1,...,l_R}$, отримаємо шукане P_k значення. Кількість методів, за допомогою яких можна здійснити розбиття k ЕМ P-програмами $N_{k,R}$, визначаємо за рекурентною формулою:

$$N_{k,R} = \sum_{j=1}^{R} \sum_{i=1}^{j} N_{k-j,i}. \qquad (4)$$

за таких початкових умов: $N_{k,1} = 1; N_{k,k} = 1; N_{k,j} = 0; k < j.$

Вірогідність того, що в ОСС є рівно k гілок за певного розподілу $l_n, l'_n (n = 1, R)$ $P^{l_1,...,l_R,l'_1,...,l'_R}$, знаходимо через розв'язок системи лінійних рівнянь, що складені за допомогою методу теорії масового обслуговування. Вірогідність зайнятості k ЕМ ОСС у момент часу $t + \Delta t$ визначаємо як суму вірогідностей таких несумісних подій:

1) у момент t k ЕМ зайняті розбиттям l_n і за час Δt не отримано жодної P-програми й жодна підсистема не завершила оброблення;

2) у момент t зайнято k-n ЕМ, проте за час Δt отримано P-програму з рангом n, $n = \overline{1,R}$,

3) у момент t зайнято $k+n$ ЕМ, але за час Δt завершилося обслуговування P-програми з рангом n, $n=\overline{1,R}$.

Перейшовши з однієї межі на іншу, якщо $\Delta t \to 0$, а потім якщо $t \to \infty$, отримуємо систему лінійних алгебраїчних рівнянь щодо шуканої вірогідності $P^{l_1,...,l_R,l`_1,...,l`_R}$. Здійснивши розв'язок рівності, обчислюємо вірогідність P_k, підсумовуючи $P_k^{l_1,...,l_R,l`_1,...,l`_R}$ за довільними комбінаціями . $l_1,...,l_R,l`_1,...,l`_R$. Використовуючи вірогідності $R_k, k=0,\infty$, визначаємо вірогідність відмов у обслуговуванні, а також вірогідність того, що всі ЕМ вільні, середнє значення зайнятих ЕМ та інші характеристики системи.

Розробимо новий підхід до аналізу функціонування СС. Передбачаємо, що ця система має різні за критеріями ресурси, що потребує різних їх комбінацій. Відповідно, пропонуємо детальну модель системи, яку назвемо БСМО.

Модель БСМО повинна ґрунтуватися на таких положеннях [12, 13]:

1. Для вузла опрацювання поставлена вимога наявності ресурсів різних типів. Хоч у самій системі їх кількість може бути довільною, проте необхідна фіксована кількість елементів кожного типу. Заявки, які надійшли, повинні опрацьовуватися з одночасним використанням різних комбінацій ресурсів, що є у системі.

2. Робота повинна належати деякому з класів і мати зафіксовані вимоги щодо ресурсів у будь-який довільний момент часу.

3. Різноманіття станів опрацювання повинно забезпечуватися класом роботи і вимогами щодо ресурсів, які зафіксовані.

4. Для будь-якого стану опрацювання необхідно визначати розподіл тривалості (часу) обслуговування між переходами з одного стану в інший.

5. Наприкінці опрацювання деякого стану щодо оброблення задачі необхідно вибирати черговий стан, що відповідає перехідній матриці вірогідностей марковського процесу.

6. Заявки, що надходять у систему, повинні мати не менше від одного джерела (необмежена кількість).

7. Необхідно визначати середню частоту потрапляння у систему робіт із довільними початковими станами опрацювання.

У системі можуть бути такі типи розподілу ресурсів, що надходять:

1) без розподілу (мультиплексування ресурсів);

2) з розподілом ресурсів;

3) мультиплексування.

Розподіл ресурсів істотно впливає на швидкість виконання робіт (її тривалість), а також на ступінь паралельного опрацювання задач у системі.

Для проведення подальшого аналізу введемо такі такі позначення:

K – кількість можливих ресурсів у системі $\overline{V_k}$;

J – кількість класів заявок;

R_i – кількість ресурсів i-го типу в системі $i=\overline{1,I}$;

I – кількість типів ресурсів;

$\overline{V_k} = \begin{pmatrix} V_{1,k} \\ \dots \\ V_{1,R} \end{pmatrix}$ – вектор вимог ресурсів на роботи, де $V_{i,k}$ – кількість необхідних ресурсів i-го типу $i=\overline{1,I}, k=\overline{1,K}$.

Стан опрацювання задачі подамо парою (j,k), яка означає: робота системи належить до класу j і потребує довільної кількості ресурсів, що надається вектором $\overline{V_k}$. Щоб спростити позначення, наводимо ізольоване подання множини пар {(j,k)} на множину {l} простих індексів;

$L = J \times K$ – кількість станів опрацювання задач;

S_l – стан l опрацювання задач $1 \le l \le L$;

$\overline{W_l} = \begin{pmatrix} w_{1,l} \\ \dots \\ w_{1,L} \end{pmatrix}$ – вектор вимог на ресурси роботи в стані S_l, де $W_{i,l}$ – кількість ресурсів i-го типу, що необхідні для довільної задачі в стані S_l .

Кінець опрацювання ототожнюється із переходом у завершений стан “0”, який позначається як S_0. Тривалість обслуговування, яке необхідне для переходу в стан S_l, – це час, упродовж якого робота займає ресурси W_l до переходу в наступний стан. Позначимо як T_l середню тривалість опрацювання довільної задачі у стані $S_l, l = \overline{1,L}$.

Щоб отримати розв’язок цієї задачі, запропонуємо судження перше. Коли необхідна робота опиняється у стані S_l, вона залишається у ньому до завершення часу її опрацювання. Судження друге. Черговий стан роботи системи наводиться дискретним марковським процесом, який описується матрицею $\overline{P}; \overline{P} = (L+1) \times (L+1)$ – це матриця перехідних вірогідностей. Оскільки заявки на роботу можуть надходити з необмеженої кількості джерел і частота їх потрапляння з початковим станом S_l дорівнює λ_l, а загальна частота (інтенсивність) $\lambda = \sum_{l=1}^{L} \lambda_l$, і, якщо буде відоме значення λ, маємо можливість обчислити вірогідність знаходження роботи у початковому її стані S_l : $f_1 = \frac{\lambda_l}{\lambda}$.

Позначимо розподіл початкових станів вектором $\overline{F}$ - [$f_1, f_2, ..., f_L$]. За допомогою вектора $\overline{F}$ і матриці $\overline{P}$ можна обчислити $\overline{G}$ - [$g_1, g_2, ..., g_L$], де g_L – середня кількість робіт, що перебувають у фіксованому стані S_l .

У багатьох БСМО передбачено наявність сталих розподілів часу опрацювання та частоти потрапляння для кожного класу заявок [14]. Розраховується $p_0(\lambda)$ – вірогідність того, що система вільна, $p_m(\lambda)$ – вірогідність того, що здійснюється обслуговування довільної комбінації m, $m = \overline{1,M}$.

Судження третє. Коли параметри $g_L, l = \overline{1,L}$ і розподіл тривалостей опрацювання є сталими, то загальна частота (інтенсивність) потоку на вході теж незмінна. З цього судження зробимо висновок, що досліджувана система має сталі характеристики потоків робіт. Тому для цього алгоритму розподілу задач і сталих характеристик потоку обчислюємо потужність системи, що виражається як інтенсивність (частота) вхідного потоку λ_{bx}: $\lim p_0(x) = 0, \lambda \to \lambda_{bx}$, де $\lambda \to \lambda_{bx}$ означає, що “λ прямує до λ_{bx} знизу”.

Межа потужності $\lambda_{\max}$ для БСМО обчислюється як інфімум інтенсивності потоків на вході, за яких насичення гарантоване, причому воно досягається незалежно від використ-

тання алгоритму розподілу заявок. Щоб визначити межі потужності БСМО, пропонуємо такий метод. Припускаємо, що g_L (середня кількість перебування роботи в стані S_l) і T_L (середня тривалість опрацювання роботи в стані S_l) є обмеженими і додатними для усіх станів опрацювання задач S_l. Тоді межа потужності $\lambda_{\max}$ є розв'язком запропонованої задачі лінійного програмування (ЛП):

$$\lambda_{\max} = \max_{p(\lambda)} \sum_{m=1}^{M} C_m p_m(\lambda), \tag{5}$$

де $\sum_{m=1}^{M} p_m(\lambda) = 1$, де $p_m(\lambda) \geq 0$, $1 \leq m \leq M$ і $\sum_{m=1}^{M} A_{Lm}(\lambda) = 0$

для $1 \leq l \leq L$, де $C_m = \dfrac{\{\sum_{l=1}^{L} e_{m,l} k_{l,m}\}}{\{\sum_{l=1}^{L} g_l T_l\}}$, $m = \overline{1, M}$; $A_{l,m} = \dfrac{e_{m,l} k_{l,m}}{\{g_l T_l - \dfrac{e_{m,l+1} k_{l+1,m}}{g_{l+1} T_{l+1}}\}}$, $m = \overline{1, M}$, $l = \overline{1, L-1}$,

де $k_{l,m}$ – кількість задач у стані S_l з комбінації m; $e_{m,l}$ – середня швидкість опрацювання задачі в стані S_l з комбінації m.

За допомогою запропонованого методу можна визначати межі потужності системи, яка має сталі характеристики потоку на вході. Оскільки у задачі ЛП є обмеження на L, то, відповідно, дорівнювати 0 повинні не більше ніж L змінних $[p_m(\lambda)]$. В цьому випадку $p_m(\lambda)$ визначає частку часу, упродовж якого система повинна опрацьовувати комбінацію m, так, щоб досягалася межа потужності M. Єдиність розв'язку цієї задачі не гарантовано, оскільки, якщо є декілька екстремальних точок, то усі опуклі їх комбінації теж будуть екстремальними.

Якщо під час проведення аналізу відомий набір $\{p_m(\lambda)\}$, на якому досягається екстремум, то можна визначити "не потрібні" для цього призначення комбінації, а саме ті, у яких $p_m(\lambda) = 0$ і "потрібні", для яких $p_m(\lambda) > 0$. Якщо ж екстремум прямує до видовження повної потужності, необхідно надавати перевагу саме "потрібним" комбінаціям. Зауважимо, що техніку отримання межі потужності "потрібних" і "не потрібних" комбінацій на практиці можна застосовувати у випадку, коли кількість можливих комбінацій невелика. Такий випадок можна застосовувати для аналізу малих систем.

Коли досліджується ОСС, яка опрацьовує множину скінченних задач, потрібно визначити достатню кількість джерел запиту ($C_{\min}$), що забезпечуватимуть результативне застосування робочого поля ОС, орієнтованого на паралельне обчислення. Воно містить множину однакових компонентів – зон, а сама система міститиме S зон. Щоб визначити $C_{\min}(d)$, запропонуємо таку модель. Припустимо, що величини запитів l_i від i-го джерела матимуть однаковий розподіл, тобто $l_i = k$ з вірогідністю P_k для усіх значень цілочислових k і $i = \overline{1, x}$. Тоді на усіх тактах розподіл ресурсів відбуватиметься за таким алгоритмом: у рандомній послідовності нумерують джерела запитів і ресурси розподілятимуться відповідно до нумерації цього такту послідовно. Відповідно, процес розподілу ресурсів буде завершеним у двох випадках: 1) якщо задоволено усі запити; 2) коли не вистачає ресурсів наступному джерелу запитів. Якщо процес розподілу ресурсів закінчився, то чергові джерела запитів будуть не розглянуті, хоч серед них могли бути такі, кількість запитів від яких не перевищувала б розміри процесорних зон, що залишилися. Метою цього опрацювання можуть бути дискретні рандомні блукання частинки

відрізком цілочислової довжини $[0,d]$. У цьому положенні (початковому) частинка перебуватиме на лівій межі відрізка, а після надання необхідного ресурсу $l_i = k$ наступного джерела запитів частинка запитів зсуватиметься на k одиниць праворуч до іншої межі відрізка. Кількість її цілих стрибків на відрізку $[0,d]$ відповідатиме кількості джерел запитів, які отримав у цьому такті роботи необхідний ресурс.

Математичне очікування l_z, за якого $d = a - Z, 0 \le z \le d < a$, кількості опрацьованих упродовж одного такту джерел запитів розподілу ресурсів можна визначити за таким алгоритмом дій. Нехай $d_{z,n}$ – вірогідність того, що за першим (початковим) положенням частинки $(0 < z < a)$ її блукання закінчується рівно через n кроків і

$$d_{z,n+1} = \sum_{x=1}^{a-1} d_{x,n} P_{x-z}. \qquad (6)$$

Визначимо, що $d_z(s) = \sum_{n=0}^{\infty} d_{z,n} S^n$. Помноживши попередні рівняння на S^{n+1}, підсумувавши за $n = 0,1,2,...,\infty$ та продиференціювавши отримане рівняння з урахуванням того, що $d_z(1) = e_z$, одержимо для $S=1$ $e_z - \sum_{x=1}^{a-1} P_{x-z} e_z = 1, \quad x = 1,2,...,a-1.$

Розв'язання цього співвідношення визначить найменшу потрібну кількість джерел запитів $C_{\min} = [e_z], \ z = 1,2,...,a-1,$ за заданої кількості процесорних зон робочого поля $d = a - z$. Якщо розподіл ресурсів здійснюється потактово, запропонований алгоритм дій може використовуватися для вибору мультипрограмування як для БП (розподіл процесорів між задачами), так і для мультипрограмних ОСС (розподіл ОП між задачами).

Запропонуємо методику, що уможливлює розрахунок варіантів розташування завдань в ОС у межах системи чи множини структур. Значення F_n структур системи за порядком n можна визначити як кількість розбиттів P_n числа n. Під структурою розумітимемо загальну кількість $A = \{l_i^{a_i}\}$ обчислювальних компонентів (ОК), де l_i – ранг ОК; a_i – кількість ОК рангів l_i; $\sum l_i a_i$ є порядком ОС; $\sum a_i = S$ – кількість ОК у системі (розмірність системи). Наведемо відповідні співвідношення (за $n < 14$):

1) кількість варіацій розміщення задач у ОС порядку n, що містить S ОК рангів $l_i < k : S_1 = d^{(k)}(n,s)$, де $d^{(k)}(n,s)$ – число Стірлінга другого роду [15];

2) кількість варіацій розміщення задач у ОС порядку n, що містить S ОК рангів $l_i \le k$:

$$S_2 = \sum_{z=0}^{n} (n) \sum_{j=0}^{S} (-1)^j d^{(1)}(n-r, s-j) d^{(k+1)}(r,j); \qquad (7)$$

3) кількість варіацій розміщення задач ОС порядку n, що містить S ОК рангів l_i;

$$S_3 = \begin{cases} 2^{-s} \sum_{i=1}^{s} (-1)^{S-1} \{i!(S-i)!\} (2i-S)^n, \frac{2}{l_i}; \\ \sum_{i=1}^{s} (-1)^{S-j} \{(S-i)!\, j!\}^1 2^{-j} \sum_{i=0}^{j} \binom{j}{i} (2i-j)^n, \frac{2}{l_i}; \end{cases} \qquad (8)$$

4) кількість варіацій розміщення задач у ОС порядку n, що містить S ОК рангів l_i, $k \le l_i \le l$;;

$$S_4 = \sum_{j=0}^{S} \sum_{r=0}^{n} (-1)^j \left(\frac{n}{r}\right) \delta^{(k)}(n-r, s-1)\delta^{(l+1)}(r, j). \quad (9)$$

Як бачимо, розрахунок кількості варіантів розміщення задач для систем об'ємного розміру становить значну обчислювальну складність, тому це частково зменшує практичну значущість запропонованої методики. Як варіант пропонуємо іншу методику визначення ймовірних параметрів ОСС під час пам'ятного опрацювання складних задач за різних дисциплін розподілу.

У випадку потактового розподілу припускаємо, що система містить M однакових процесорів, а кількість задач, які потрібно розв'язати під час кожного розподілу, дорівнює k; i - s, $i = \overline{T,k}$ Для розв'язання задачі з однаковою вірогідністю необхідно N_i процесів із множини $\{0,1, ..., N\}$; задачі водночас містять всі надані їм процесори, які після закінчення опрацювання одночасно звільняються; тривалість розв'язання кожної задачі не перевищує T.

Вірогідність $P_k(M,N)$ розв'язання усіх k задач за один такт визначається як відношення значення B усіх сприятливих варіантів $\sum_{i=1}^{k} n_i \le M$, $0 < n_i < N$, $i = \overline{1,k}$ до значення W всіх можливих варіантів $0 \le n_i \le N$, $i = \overline{1,k}$.

Отже, $P_k(M,N) = \dfrac{B}{W} = \dfrac{\sum_{n_1=0}^{M} \sum_{n_2=0}^{M-n_1} \cdots \sum_{n_k=0}^{M-n_1-\ldots-n_{k-1}} \prod_{i=1}^{k} h_i}{(N+1)^k}$, де $h_i = \begin{cases} 1, & 0 \le h_i \le N, \\ 0, & h_i > N. \end{cases}$

Наведена рівність не забезпечує на практиці знаходження вірогідності $P_k(M,N)$ за доволі великих значень M, N і k. Тому отримано співвідношення, що дає наближене значення $P_k(M,N)$:

$$P_k(M,N) = \frac{1}{k!} \sum_{j=0}^{\left[\frac{M}{N}\right]} (-1)^j C_k^j \left(\frac{M}{N} - j\right)^k + 0\left(\frac{2}{N}\right), \quad (10)$$

де [ ] – ціла частина числа.

Воно придатне для практичного розв'язання завдань і справедливе за $M >> k$ і $N >> k$. Відповідні параметричні співвідношення придатні для проєктування сучасних ОСС, що повинні складатися із великої кількості мініпроцесорів і забезпечувати безперебійну роботу сотням і тисячам користувачів [16]. Застосовуючи $P_k(M,N)$, визначимо розподіл вірогідності кількості розв'язаних за такт задач:

$$P_{k,S}(M,N) = P_S(M,N) \prod_{i=1}^{k-S} [1 - P_{S+i}(M,N)], \quad S \le k; \quad (11)$$

а математичне очікування розв'язаних задач за такт:

$$\bar{k}(k,M,N) = \sum_{S=1}^{k} S P_{k,S}(M,N), \quad (12)$$

та інші характеристики ОСС. На рис. 2–4 наведено графіки залежності $P_k(M,N)$, $\bar{k}(k,M,N)_n$, $\dfrac{\bar{k}(k,M,N)}{k}$ за різних значень M, N і k.

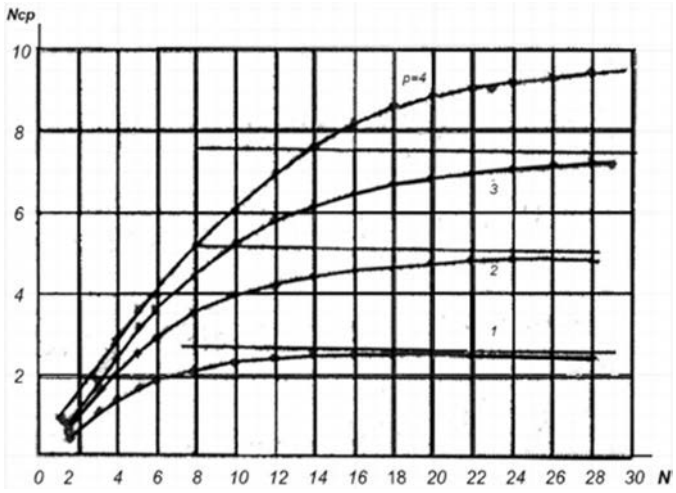

Рис. 1. Залежність середньої кількості загальних процесів N_{cp} *від загальної кількості* N *ЕМ ОСС у разі обслуговування із чергою у різноманітних значеннях* ($a_n = 0{,}25;\ n = 1{,}4$)

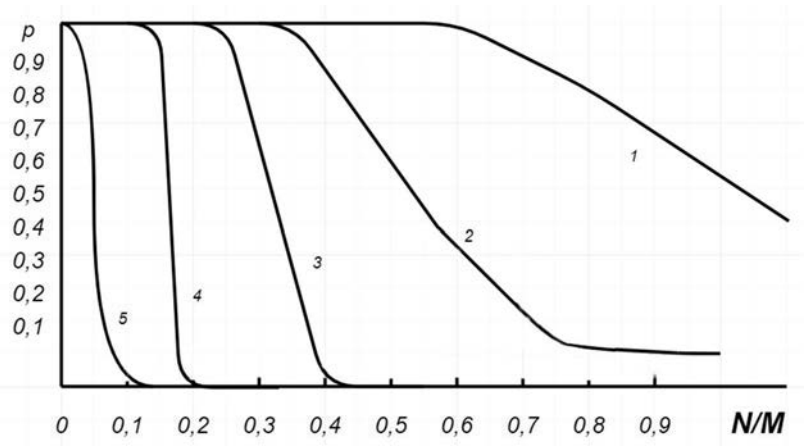

Рис. 2. Графік залежності $P_k(M, N)$ *від відношення* $\frac{N}{M}$, *якщо k дорівнює: 2 (крива 1), 4 (крива 2), 8 (крива 3), 16 (крива 4) і 32 (крива 5)*

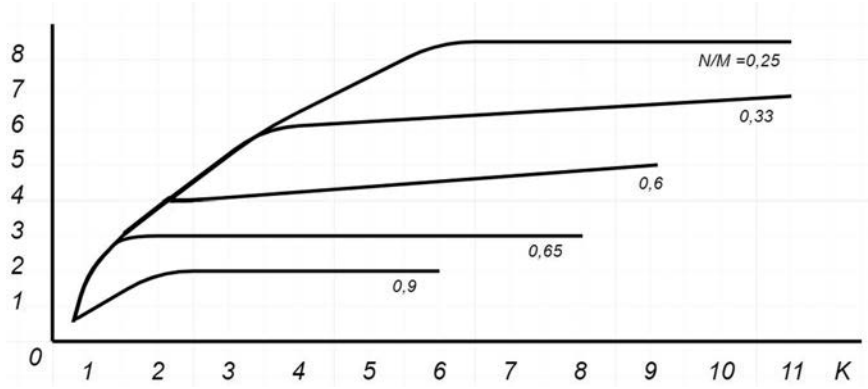

Рис. 3. Графік залежності математичного очікування кількості опрацьованих заявок $\bar{k}$ *від загальної кількості користувачів k за різних значень відношення* $\frac{N}{M}$

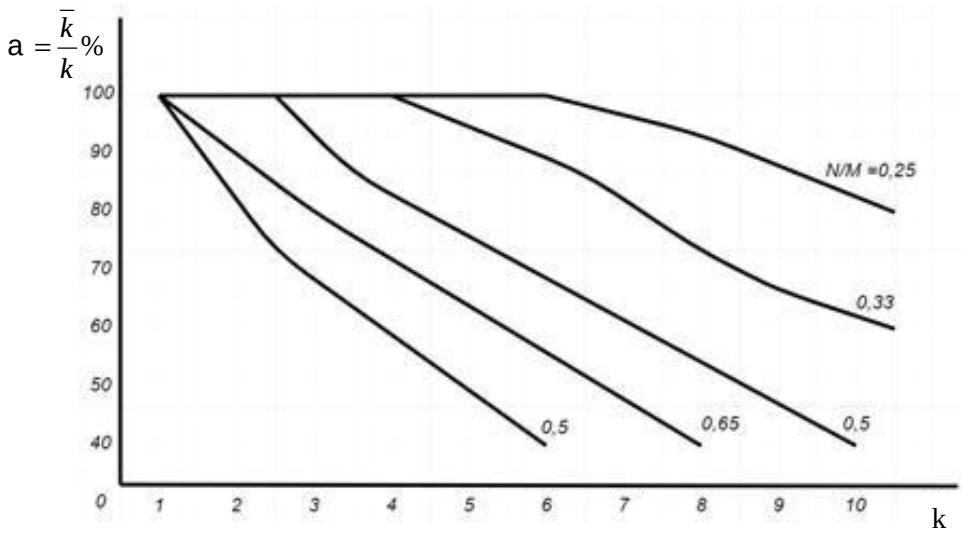

Рис. 4. Залежність відношення $a = \frac{\bar{k}}{k}$ *від загальної кількості користувачів k за різних значень відношення* $\frac{N}{M}$

Висновок

Отже, результати, отримані в роботі, ототожнюються з поведінкою систем та з насиченням системи у разі зростання кількості користувачів. Крім того, виявлено вірогідність розв'язання усіх задач системою, що дає змогу прогнозувати продуктивність ОСС. Аналізування математичних моделей щодо функціонування СС допомагає здійснювати вибір тієї чи іншої системи для розв'язання конкретної задачі.

Результати можуть бути використані під час вибору параметрів систем, які проєктуються.

Список літератури

1. Swan R. S., Fuller S. H., Siewiorek D. P. (2007). Modular Multimicroprocessor. *AFIPS Couf. Proc.*, Montrale N.Y. r. 46, 637–644.

2. Ларичев О. И. (2002). Теория и методы принятия решений. М.: Логос. 374 с.

3. Лихачевич В. С., Михалевич В. С., Волкович В. Л. (1982). Вычислительные методы исследования и проектирования сложных систем. М. : Наука, 286 с.

4. Згуровський М. З., Панкратова Н. Д. (2007). Основи системного аналізу. К.: Видавнича група BHV, 544 с.

5. Флинн М. Дж. (1972). Некоторые компьютерные организации и их эффективность. *IEEE Transactions on Computers*, 21(9), 948–960.

6. Богданов А. В., Корхов В. В., Мареев В. В., Станкова Е. Н. (2004). Архитектуры и топологии многопроцессорных вычислительных систем. М.: ИНТУИТ.РУ "Интернет-Университет Информационных Технологий", 176 с.

7. Михайлов Б. М., Халабия Р. Ф. (2010) Классификация и организация вычислительных систем: учеб. пособ. М.: МГУПИ, 144 с.

8. Хорошевський В. Г. (2008) Архитектура вычислительных систем: учеб. пособ. 2-е изд., перераб. и доп. М.: Изд-во МГТУ им. Н.Э. Баумана, 520 с.

9. Зубатенко В. С., Майстренко А. С., Молчанов И. Н. и др. (2006). Исследование некоторых параллельных алгоритмов решения задач линейной алгебры на MIMD-компьютерах. *Искусственный интеллект*, № 3, 129–138.

10. Яковлєв М. Ф., Нестеренко А. Н., Бруснікін В. М. (2014). Проблеми ефективного розв'язування систем нелінійних рівнянь на багатопроцесорних комп'ютерах MIMD-архітектури. *Математичні машини і системи*, № 4, 12–17.

11. Regis R. C. (2003) Multiserves Queueing Models of Multiprocessor Systems. *IEEE Trans. v. c.*, 22, No. 8, 736–744.

12. Иванченко Е. В., Спастенко Е. П., Хорошко В. А. (2013). Синтез структурно-оптимальной системы управления сложными... *Інформаційна безпека*, № 4(12), 126–130.

13. Kenneth J. O. (2007). Capacity Bounds for Multirescurse Quenes. *Journal of ACM*, vol. 24 NY, 648–663.

14. Егоров Ф. И., Орленко В. С., Хорошко В. А. (2007). Проектирование сложных зашифрованных сетей. *Вісник ДУІКТ*, Т. 5, № 4, 2007, 39–51.

15. Оре О. (1980). Теория графов. М.: Нация, 1980, 338 с.

16. Брайловский Н. Н., Хорошко В. А. (2014). Оптимизация характеристик сложных систем по критерию живучести. *Інформаційна безпека*, № 1(13), 17–32.

References

1. Swan R. S., Fuller S. H., Siewiorek D. P. (2007). Modular Multimicroprocessor. AFIPS Couf. Proc., Montrale N. Y. V. 46, 637–644.

2. Larichev O. (2002). Theory and methods of decision making. M.: Logos, 374 p.

3. Likhachevich V., Mikhalevich V., Volkovich V. (1982). Computational methods for research and design of complex systems. M.: Science, 286 p.

4. Zgurovsky M., Pankratova N. (2007). Fundamentals of system analysis. K.: BHV Publishing Group, 544 p.

5. Flynn M. (1972). Some computer organizations and their effectiveness. *IEEE Transactions on Computers*. No. 21 (9), 948–960.

6. Bogdanov A., Korkhov V., Mareev V., Stankova E. (2004). Architectures and topologies of multiprocessor computing systems. – M .: INTUIT.RU "Internet University of Information Technologies", 176 p.

7. Mikhailov B., Khalabiya R. (2010). Classification and Organization of Computing Systems: A tutorial. M. : MGUPI, 144 p.

8. Horoshevsky V. (2008). Architecture of Computing Systems: A Textbook. allowance. 2nd ed., Revised. and ext. M .: Publishing House of the Moscow State Technical University H. E. Bauman, 520 p.

9. Zubatenko V., Maistrenko A., Molchanov I. And others (2006). Investigation of some parallel algorithms for solving linear algebra problems by MIMD computers. *Artificial intelligence*, No. 3, 129–138.

10. Yakovlev M., Nesterenko A., Brusnikin V. (2014). Problems of effective solution of systems of nonlinear equations on multiprocessor computers MIMD-architecture. *Mathematical Machines and Systems*, No. 4, 12–17.

11. Regis R. C. (2003). Multiserves Queueing Models of Multiprocessor Systems. *IEEE Trans.v.c.*, 22, No. 8, 736–744.

12. Ivanchenko E., Spastenko E., Khoroshko V. (2013). Synthesis of structure-optimal from complex control system. *Information Security*, No. 4 (12), 126–130.

13. Kenneth J. O. (2007). Capacity Bounds for Multirescurse Quenes. *Journal of ACM*. V. 24 NY, 648–663.

14. Egorov F., Orlenko V., Khoroshko V. (2007). Designing Complex Encrypted Networks. *DWICT Bulletin*, vol. 5, No. 4, 39–51.

15. Ore O. (1980). Graph theory. M.: Nation, 338 p.

16. Brailovsky N., Khoroshko V. (2014). Optimization of the characteristics of complex systems by the criterion of survivability. *Information Security*, No. 1 (13), 17–32.

ANALYSIS OF MATHEMATICAL MODELS OF FUNCTIONING SCALAR MULTIPROCESSOR SYSTEMS

Ivan Opirskyy[1], Serhii Zybin[2], Vladimir Khoroshko[3]

[1] Lviv Polytechnic National University, Department of Information Protection
iopirsky@gmail.com, ORCID: 0000-0002-8461-8996
[2, 3] National Aviation University, Department of Information Technology Security
[2] professor_va@ukr.net, ORCID: 0000-0001-6213-7086
[3] duikt@ua.fm, ORCID: 0000-0002-2670-2823

The article is devoted to the research of mathematical models of functioning of complex systems. The correspondence of the results of the study with the qualitative idea about the behavior of the system, saturation of the system with increasing number of users.

A model of a homogeneous complex system with priority (flow) of large tasks is constructed. The studies of the functioning of a homogeneous complex system in processing the flows of complex tasks have been carried out on the assumption that a Poisson flow of programs (P-programs) with a certain frequency enters the system of *N* elementary personal electronic computers with additional infinite external memory. New approach to the study of the functioning of complex systems is proposed. The system is assumed to have different types of resources and applications that require different combinations of these resources. Thus, the paper proposes a rather detailed model of the system, which is called a multi-resource system of massive maintenance. A technique for determining the power limit of a multi-resource mass service system is proposed. This technique allows you to determine the power limits for a system with fixed input stream characteristics. The model is proposed that allows to establish the minimum required number of sources of queries at a given number of processing zones of the work field. When resources are redistributed tactfully, the model described can be used to select multiprogramming for both multiprocessor (processor allocation between tasks) and multiprogram single-processor computing systems (allocation of RAM between tasks).

The method of calculating the number of variants of assignments of tasks in a computer system within the given structure or set of structures, as well as the method of determining the probable characteristics of homogeneous complex systems in the mode of memorizing the processing of complex problems in different disciplines of distribution are offered. It is suitable for the design characteristics of modern homogeneous complex systems, which should contain hundreds and thousands of mini-processors and serve hundreds of users.

The analysis of mathematical models concerning the functioning of complex systems is carried out, which allows the choice of a particular system to solve a specific problem. The results obtained are consistent with the perceptions of system behavior and system saturation as the number of users increases. In addition, the probability of solving all problems by the system is revealed, which allows to predict the performance of homogeneous complex systems and to be used in choosing the parameters of the designed systems.

Key words: multiprocessor system, homogeneous complex system, multi-resource queuing system, resource allocation process, processing flows, probability, rank programs, power limit.