

Ю. С. Клушин

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

ПІДВИЩЕННЯ ТОЧНОСТІ ОЦІНЮВАННЯ ЧАСУ ВИКОНАННЯ СКЛАДАНИХ ПРОГРАМНИХ КОМПЛЕКСІВ У БАГАТОПРОЦЕСОРНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ ПІД ЧАС ПОЯРУСНОГО СТОХАСТИЧНОГО МОДЕЛЮВАННЯ

© Клушин Ю. С., 2017

Для підвищення точності оцінювання часу виконання складних програмних комплексів на паралельних комп'ютерах розроблено алгоритм рівномірного розподілу вершин графу заданого комплексу взаємопов'язаних робіт. Цей алгоритм використовується у методі поярусного стохастичного моделювання у багатопроцесорних комп'ютерних системах.

Ключові слова: паралельні обчислювальні системи, комплекс взаємопов'язаних робіт, пряме стохастичне моделювання, марковський процес.

Y. Klushyn

Lviv Polytechnic National University,
Computer Engineering Department

INCREASING THE ACCURACY OF EVALUATION OF THE PERFORMANCE PERIOD OF SOFTWARE COMPLEX COMPONENTS IN MULTIPROCESSOR COMPUTER SYSTEMS UNDER NOISE STOCHASTIC MODELING

© Klushyn Y., 2017

To improve the accuracy of the estimation of the time of execution of complex software complexes on parallel computers, an algorithm for the uniform distribution of vertices of a graph of a given set of interrelated works is developed. This algorithm is used in the method of plywood stochastic simulation of multiprocessor computer systems.

Key words: parallel computing systems, complex interconnected works, direct stochastic modeling, Markov process.

Вступ

Сьогодні через істотне зростання бізнесу, необхідність використання корпоративних систем, оброблення великого обсягу інформації, недопущення збоїв і простоювань у роботі у комп'ютерних системах застосовується принцип багатопроцесорної архітектури, який передбачає спільну роботу декількох процесорів одночасно, що значно збільшує продуктивність комп'ютерної системи. Кількість процесорів, використовуваних у системі, залежить не тільки від потужності обчислювальних машин, але і від розв'язуваних ними завдань, і може варіюватися від двох–чотирьох до декількох десятків. Цей принцип архітектури почали застосовувати і в таких сферах, як оброблення транзакцій у режимі реального часу, створення сховищ даних, для організації систем

підтримки прийняття рішень, управління різноманітними об'єктами в реальному часі. Завдяки цьому очевидної актуальності набуває проблема апріорної оцінки "придатності" таких паралельних обчислювальних систем (ОС) для розв'язання конкретного набору задач, що задає користувач, за необхідний час.

Стосовно паралельних ОС ця проблема отримала назву "прогнозування часу виконання складних програмних комплексів"; останні зазвичай задають графовими моделями і розглядають як *комплекси взаємопов'язаних робіт* (КВР) – задач і / або їх паралельно-последовних фрагментів (підзадач, процесів, програмних модулів).

Формально під прогнозуванням часу виконання конкретного комплексу робіт розуміють стохастичну оцінку часу T його реалізації (середнє значення, дисперсія, функція розподілу T) на паралельній ОС, а також визначення ймовірності виконання комплексу за час, не більший від заданого "директивного" часу T_{\max} [1–3].

Важливо зазначити, що час виконання кожного програмного модуля (роботи) і КВР загалом тут розглядають як випадкову величину – через довільну кількість логічних розгалужень у програмі модуля, циклів невизначеної довжини, випадкового характеру взаємодії між процесами та звернень до зовнішньої пам'яті, конфліктів на загальних ресурсах паралельних ОС тощо. Тому використання відомих точних методів оцінювання T , наприклад, методів теорії розкладів, для вирішення проблеми прогнозування у зазначеній постановці неприйнятне.

Аналіз останніх досліджень та публікацій

Розроблення точних математичних моделей і алгоритмів для аналізу функціонування паралельних ОС на КВР, які задає користувач, з випадковим часом виконання кожної роботи (процесу) дозволило б розв'язати актуальну задачу достовірної аналітичної оцінки часу виконання кожного конкретного КВР на ОС апріорно до вибору структури і конфігурації паралельної ОС або до детального розроблення програм КВР.

Можна виділити такі основні підходи до оцінювання ефективності ОС у зазначеному сенсі: пряме або непряме вимірювання часу виконання реальних програм заданого набору на ОС, імітаційне та математичне моделювання.

Експериментальне дослідження ефективності ОС здійснюється безпосереднім вимірюванням значень показників системи, що працює. Завантаження ОС або її пристроїв можна забезпечити реальними програмами, типовими для заданої області застосування, або за допомогою спеціальних трас, які повинні, за можливості, повно відображати особливості реального завантаження ОС [4].

Імітаційне (статистичне) моделювання зазвичай використовується для порівняльного аналізу альтернативних архітектурних і структурних рішень під час проектування різних вузлів ОС, а також для оцінювання точності математичного моделювання. Питання створення і застосування методів імітаційного моделювання доволі добре висвітлено в літературі (див., наприклад, [5–7]). Перевагою імітаційного моделювання є можливість аналізу роботи ОС практично з будь-яким ступенем деталізації. Однак дослідження ОС методами імітаційного моделювання у загальному випадку – трудомісткий і складний процес. Витрати на створення імітаційних моделей і проведення моделювання, навіть з використанням спеціалізованих мов, доволі великі.

Для дослідження ефективного функціонування і поведінки паралельних ОС на етапі їх проектування або вибору їх структури і конфігурації для передбачуваної області застосування найчастіше використовується математичне моделювання. Розрізняють детерміновані й стохастичні (ймовірні) моделі. Детерміновані моделі застосовують для оцінювання значень порівняно простих параметрів продуктивності ОС і тому можливість їх використання для оцінювання ефективності ОС вельми обмежена.

У дослідженнях ОС стохастичні математичні моделі застосовують значно ширше. Ці моделі ґрунтуються на математичному апараті марковських ланцюгів і теорії систем масового обслуговування. Перевагою аналітичних моделей ОС є достатня спільність і наочність отриманих результатів і порівняно невелика трудомісткість побудови і дослідження моделей. ОС розглянутого тут класу – однорідні багатопроцесорні системи з багатьма потоками команд і даних (БКБД) – легко інтерпретуються як системи масового обслуговування унаслідок таких факторів;

- паралельні функціональні блоки ОС можна трактувати як задану кількість паралельних обслуговуючих пристроїв (процесорів, пристроїв введення–виведення тощо);
- звернення одного пристрою до іншого відбуваються асинхронно, у випадкові моменти часу; на входах обслуговуючих приладів формуються черги з випадковою кількістю заявок.

Метод поярусного стохастичного моделювання

Одним з таких методів, який найточніше вирішує проблему прогнозування часу виконання КВР, є метод прямого стохастичного моделювання [3, 4, 8]. Розвитком цього методу є метод поярусного стохастичного моделювання [9], який зменшує кількість станів марковського процесу з незначним (на одиниці відсотків) збільшенням похибки прогнозування виконання КВР у паралельних ОС.

Розглянемо суть методу поярусного стохастичного моделювання.

Граф G заданого КВР (рис. 1) описується таблицею зв'язності його вершин (табл. 1). Таблиця містить N рядків (за кількістю вершин графу G), в кожній з яких вказано номери робіт (вершин), що є попередниками і наступниками цієї роботи.

Процес виконання КВР представлено математичною моделлю (рис. 2) у вигляді однофазної СМО з $k \geq 2$ однотипних ОП з буфером B для готових до виконання робіт (поточний фронт F); останні надходять з пулу Π , що містить у початковому стані N робіт. Вершини графу G (A, H), який визначає заданий КВР, розділяють за кількістю вершин на шари (яруси), кількість яких визначається кількістю вершин критичного шляху в цьому графі (рис. 1).

Кожен шар являє собою безліч паралельних робіт, які можуть виконуватися на ОП одночасно.

Зауважимо, що розбиття вершин графу по ярусах може бути неоднозначним.

Для математичної моделі процесу $X(t)$ це означає, що пул робіт Π розділяється за кількістю шарів на R пулів, де R – довжина критичного шляху в графі КВР.

Модель функціонує у безперервному часі. Випадковий час t_j обслуговування будь-якої роботи a_j вважатимемо розподілений за експоненціальним законом із параметром $m_j = 1/M[t_j]$, але в нашому випадку значення j залишаються різними для різних робіт залежно від конкретних значень $M[t_j]$.

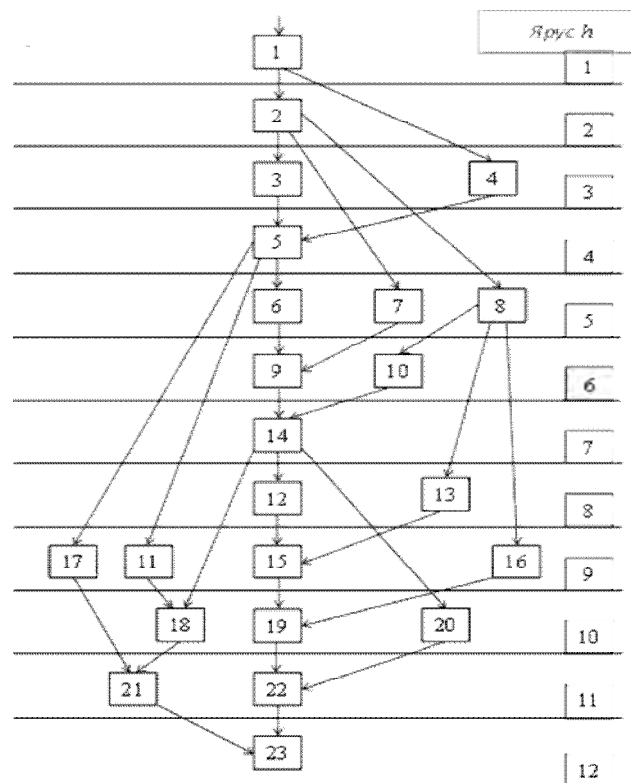


Рис. 1. Граф G заданого КВР

У початковий момент часу в систему (буфер **Б** і обслуговувальні прилади **ОП**) надходить з пулу **П** одна заявка – “початкова” робота a_1 , яка негайно починає обслуговуватися на одному з **ОП**. Після завершення обслуговування a_1 (у загальному випадку a_j) в **ОП** ця робота залишає систему, “передаючи” свій номер j в пул **П**, з якого вибирають у буфер **Б** ті наступники роботи a_j , які виявилися готовими до виконання (тобто виконані всі їхні попередники); номери цих робіт однозначно встановлюються за таблицею зв’язності вершин графу G і належать деякій підмножині шарів від h по $h + l - 1$ (“вікно перегляду”), причому:

- значення h може змінюватися від 1 до R ;
- l – кількість шарів у “вікні перегляду” може змінюватися від 1 до R ;
- h – номер початкового шару в “вікні перегляду”;
- $h + l - 1$ – номер кінцевого шару у “вікні перегляду”.

Отже, в кожен момент часу в системі можуть перебувати не всі роботи КВР, готові до виконання (як в [3]), а тільки ті з них, які є у “вікні перегляду” з l сусідніх ярусів, причому це “вікно” переміщається по графу КВР на один ярус, як тільки завершено виконання робіт шару h .

Зауважимо, що якщо $l = R$, поярусне моделювання зводиться до прямого стохастичного моделювання виконання КВР загалом [3].

Таблиця 1

Таблиця зв’язності вершин графу G (рис. 1)

Номер роботи a_j	Роботи-попередники $\{a_{пп}\}_j$	Роботи-наступники $\{a_{нп}\}_j$	$M [t_j]$	μ_j	Ранг r_j	Зв’язність b_j
1	2	3	4	5	6	7
1	-	2,4	150	0,0067	12	2
2	1	3,7,8	30	0,0333	11	3
3	2	5	160	0,0063	10	1
4	1	5	80	0,0125	10	1
5	3,4	6,11,17	70	0,0143	9	3
6	5	9	120	0,0083	8	1
7	2	9	170	0,0059	8	1
8	2	10,13,16	50	0,0200	8	3
9	6,7	14	130	0,0077	7	1
10	8	14	150	0,0067	7	1
11	5	18	100	0,0100	4	1
12	14	15	160	0,0063	5	1
13	8	15	100	0,0100	5	1
14	9,10	12,18,20	40	0,0250	6	3
15	12,13	19	30	0,0333	4	1
16	8	19	170	0,0059	4	1
17	5	21	100	0,0100	3	1
18	11,14	21	70	0,0143	3	1
19	15,16	22	130	0,0077	3	1
20	14	22	80	0,0125	3	1
21	17,18	23	120	0,0083	2	1
22	19,20	23	50	0,0200	2	1
23	-	-	40	0,0250	1	0

У межах кожного “вікна перегляду” система функціонує за алгоритмом, викладеним у роботі [10].

Функціонування СМО описується марковським процесом, що обривається, $X(t)$ над безліччю станів $X = \{(m; \overset{\cdot}{i}_w; \overset{\cdot}{j}_n)\}$; $\overset{\cdot}{i}_w$ містить номери робіт, які готові до виконання і очікують у буфері Б, а $\overset{\cdot}{j}_n$ – номери робіт, які перебувають на обслуговуванні в **ОП**.

Однак тепер всі роботи, які входять в i_w^1, j_n^1 , належать шарам від h до $h + l - 1$. Це означає: поки не виконано всі роботи h -го шару, не можуть виконуватися роботи шарів, наступних за шаром $h + l - 1$.

Після виконання всіх робіт h -го шару загальна кількість шарів у “вікні перегляду” знову доводиться до l (якщо не вичерпано безліч шарів R) і алгоритм, описаний вище, повторюється. Отже, особливість функціонування СМО полягає тут у тому, що в системі в кожен момент часу можуть перебувати тільки роботи, що належать шарам з поточними номерами від h по $h + l - 1$.

Вважатимемо, що задана деяка дисципліна (критерій) K [14] для однозначного вибору робіт з буфера B на обслуговування в ОП, наприклад, вибір роботи з найменшим номером, тому в разі звільнення ОП вищий пріоритет на обслуговування мають роботи відповідно до заданого критерію K (правило диспетчеризації) [14].

Для ілюстрації функціонування моделі (рис. 2) під час виконання КВР згідно з рис. 1 приймемо, що система містить два ОП, тобто $k=2$, і у неї в кожний момент часу можуть бути тільки роботи двох шарів, тобто $l=2$.

Попередники кожної роботи a_j тут вказуються тільки в тому випадку, коли вони належать попереднім $l-1$ шарам. Наприклад, для роботи a_{11} , яка належить 9-му шару (позначається $a_{11(9)}$) у табл. 1 вказано, що вона не має попередників (хоча фактично такою є робота $a_{5(4)}$): якщо $l=2$, роботи 4-го та 9-го ярусів не можуть одночасно перебувати у “вікні перегляду” та у системі.

Аналогічно, наступників кожної роботи a_j вказують лише в тому випадку, коли вони належать наступним $l-1$ ярусам.

Для КВР (рис. 1) після закінчення обслуговування $a_{1(1)}$, що належить першому шару та пулу Π_1 , виявляються готовими до виконання роботи $a_{2(2)}$ та $a_{4(3)}$ другого та третього шарів з Π_2 та Π_3 (див. табл. 1), які надходять в систему, а у цьому випадку безпосередньо на ОП. Коли першою з цих робіт виконалася $a_{4(3)}$, то у систему не надходять нові роботи, оскільки $a_{5(5)}$ – наступник роботи $a_{4(3)}$ не готовий до виконання (не виконана робота $a_{3(3)}$), а в системі перебувають тільки роботи 2-го ярусу – $a_{2(2)}$. Якщо ж завершилося виконання $a_{2(2)}$ (єдиного у пулі Π_2), то у систему можуть надходити роботи двох наступних ярусів (якщо $l=2$), тобто з Π_3 та Π_4 , але в цьому випадку надходить тільки одна робота $a_{3(3)}$ – наступник роботи $a_{2(2)}$. Тепер у системі перебуватимуть дві роботи 3-го шару – $a_{3(3)}$ та $a_{4(3)}$. Після завершення виконання однієї з цих робіт ($a_{3(3)}$ або $a_{4(3)}$) у систему нові роботи не надходять, оскільки в обох випадках не готова до виконання робота $a_{5(4)}$. Зауважимо, що поки не виконані обидві роботи 3-го ярусу – $a_{3(3)}$ та $a_{4(3)}$ (з Π_3), у систему не будуть надходити роботи 5-го ярусу (з Π_5), зокрема $a_{7(5)}$ та $a_{8(5)}$. Для математичної моделі за рис. 4 це означає, що роботи $a_{7(5)}$ та $a_{8(5)}$ не мають попередників, робота $a_{3(3)}$ має тільки одного наступника – роботу $a_{5(4)}$ (див. табл. 1).

Алгоритм визначення стану процесу під час виконання конкретного КВР описано в роботі [9].

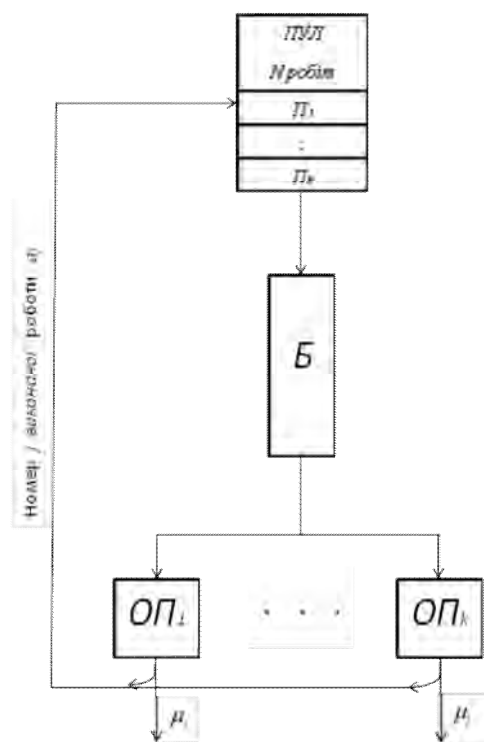


Рис. 2. Математична модель

Постановка задачі

Метою статті є аналіз залежності точності результатів методу поярусного стохастичного моделювання під час прогнозування часу виконання комплексу взаємопов'язаних робіт, який

задають графом, від розташування вершин цього графу відповідно до ярусів (шарів). На основі цього аналізу необхідно розробити алгоритм, який би дав змогу однозначно розташовувати вершини графу згідно з ярусами (шарами) за максимальної точності часу прогнозування КВР.

Алгоритм рівномірного розподілу вершин графу

Дослідження, показали, що більшої точності результатів прогнозування за допомогою поярусного стохастичного моделювання можна досягти використанням алгоритму рівномірного розподілу вершин (робіт) КВР (графу G) згідно з ярусами. Цей алгоритм ґрунтується на понятті ступеня свободи d_j вершини (роботи) a_j , яка визначається кількістю ярусів, у яких може бути розміщена (може “відвідати”) вершина a_j за ярусно-паралельного подання графу G .

Пояснимо це на прикладі. На рис. 3 наведено варіант розбиття графу G на шари, за якого вершини розташовані рівномірно вздовж критичного шляху графу. На рис. 4 усі вершини цього ж графу, які не належать його критичному шляху, відносять до одного шару з найменшим номером. У табл. 2 наведено числові значення поярусного стохастичного модулювання для визначення середнього часу \bar{T} виконання того самого КВР, але з різними розбиттями його вершин за шарами – відповідно до рис. 3 та 4 – за кількості шарів у “вікні перегляду” $l=2$ та кількості обслуговувальних приладів $k=2$. Похибка оцінок \bar{T} підраховувалась відносно результатів прямого стохастичного моделювання того ж КВР (тобто щодо $l=R^*=4$ для нашого прикладу).

Прогнозований час \bar{T} для іншого випадку, відповідного рис. 4, виявився істотно великим порівняно з прямим стохастичним моделюванням (насправді виявилась достатньо великою похибка прогнозування). Цього можна було очікувати, оскільки в цьому випадку, якщо $l=2$, робота, наприклад, $a_{6(4)}$ не влучає у “вікно перегляду” до закінчення виконання усіх робіт 2-го шару, і тільки після цього моделюватиметься виконання робіт $a_{6(4)}$, $a_{6(5)}$, $a_{10(6)}$; оскільки вони утворюють послідовний ланцюг робіт, тобто виконуються на одному ОП, це неминує призводить до погіршення оцінки \bar{T} .

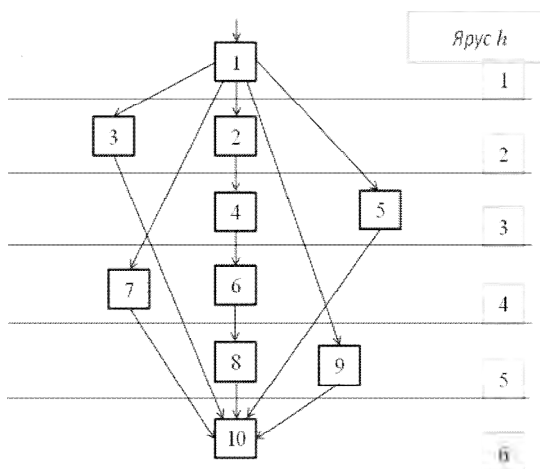


Рис. 3. Граф з рівномірним розподілом вершин

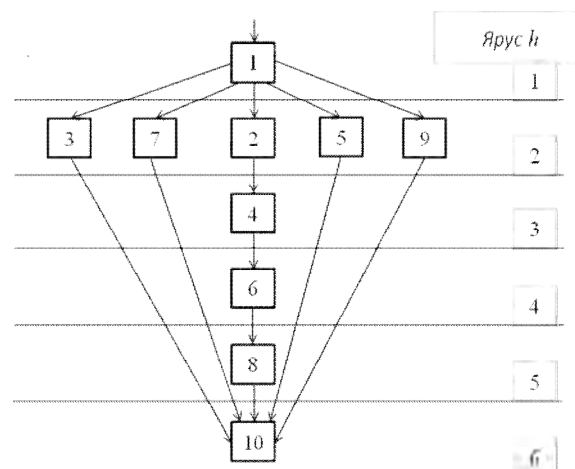


Рис. 4. Граф з вершинами, які належать мінімальному ярусу

Звідси можна зробити припущення про меншу похибку (для нашого прикладу – приблизно у чотири рази) результатів прогнозування \bar{T} поярусним стохастичним моделюванням, якщо вдатися до процедури рівномірного розподілу вершин (робіт) КВР (графу G) за шарами.

Перш ніж перейти до цієї процедури, ведемо поняття ступеня свободи вершини a_j .

Ступінь свободи d_j роботи a_j визначається кількістю шарів, у яких може бути розміщена (може відвідати) вершина a_j (зрозуміло, з урахуванням розміщення її попередників і наступників); ступінь свободи будь-якої вершини критичного шляху дорівнює 1.

Ступінь свободи d_j роботи a_j можна визначити за ярусно-паралельним поданням графу G (див., наприклад, [9]) переміщенням усіх вершин графу на яруси з максимально та мінімально можливими номерами.

Таблиця 2

Результати числового розрахунку середнього часу виконання КВР (рис. 3, рис. 4)

Метод моделювання	Кількість S станів ОМП	Середній час \bar{T}	Похибка $\Delta = \frac{T - T_{np.cm.}}{T_{np.cm.}} \cdot 100 \%$
Пряме стохастичне мод., значення $m=0,01$ (рис. 3). Критерій K : “ r_j/b_j ”	32	669	0
Поярусне стохастичне моделювання за значень $m=0,01$; $l=2$ згідно з рис. 4	25	743	11
Поярусне стохастичне моделювання за значень $m=0,01$; $l=2$ згідно з рис. 3	23	689	3
Поярусне стохастичне моделювання за значень $m=0,01$; $l=3$ згідно з рис. 3	29	672	0,4

Наприклад, для графу G (рис. 3 і 4) ступінь свободи кожної з вершин $a_1, a_2, a_4, a_6, a_8, a_{10}$ дорівнює 1 (це вершини, які належать критичному шляху), а у кожній з вершин a_3, a_7, a_5, a_9 ступінь свободи дорівнює 4. Для графу G згідно з рис. 1, який ми продовжуємо використовувати для ілюстрації підходу загалом, показано розміщення вершин за ярусами з найбільшими номерами або, що те саме, відповідно до значень їх рангів згідно з [10–13] (рис. 1).

Нижче викладено евристичний алгоритм рівномірного розподілу вершин графу КВР згідно з ярусами, оснований на використанні ступеня свободи кожної вершини.

Позначимо: A_N – безліч всіх вершин графу; A_k – підмножина вершин критичного шляху (шляхів), кожна з яких має ступінь свободи $d_k = 1$; A_{N-k} – підмножина інших вершин, у кожній з яких $d_k > 1$.

Результатом виконання алгоритму буде заповнення стовпчика 7 табл. 3 – вказується номер шару, до якого буде зарахована робота, а також стовпців 2 і 3 за заданої ширини l “вікна перегляду”.

Узагальнений опис алгоритму

1. Визначається критичний шлях (шляхи) у графі A_k – підмножина вершин. Кожній вершині a_k критичного шляху приписується значення ступеня свободи $d_k = 1$ (стовпець 6 табл. 3). Перехід до п. 2.

2. Визначається ступінь свободи d_j кожної вершини підмножини A_{N-k} щодо вершин критичного шляху (стовпець 6 табл. 3). Перехід до п. 3.

3. Для кожної вершини a_{kj} критичного шляху, в послідовності їх розміщення, формується “Список 1” вершин, які можуть бути розміщені в тому ж шарі, що і a_{kj} (тобто список робіт, які можуть виконуватися паралельно з роботою a_{kj} – стовпець 8 табл. 3). Перехід до п. 4.

4. Вибирається чергова вершина a_{kj} – критичного шляху в послідовності їх розміщення з безлічі A_k . Перехід до п. 5.

5. Чи є вибрана вершина a_{kj} останньою з безлічі A_k ?

Так. Вершина a_{kj} належить шару R . Перехід до п. 7.

Ні. Перехід до п. 6.

6. Перевіряють, чи порожній “Список 1” для вибраної вершини a_{kj} .

Ні. Зі “Списку 1” для вибраної вершини a_{kj} вибирають вершини з найменшими d_j , так щоб кількість вершин, включаючи і вершину a_{kj} , була пропорційною до кількості ОП. Це означає, що у разі першого вибору вершини a_{kj} (а значить, і шару) кількість вершин, що розміщуються у шарі, має бути не більшою за кількість ОП, якщо дозволяє “Список 1”.

Наприклад, для вибраної вершини a_{k2} , табл. 3, і якщо $k = 2$, і $k = 3$, кількість вершин у шарі дорівнюватиме 2 (вершини 2 і 4); а для a_{k3} відповідно 2 (вершини 3, 7) і 3 (вершини 3, 7, 8). У разі повторного вибору вершини a_{kj} кількість вершин, що розміщуються в шарі, має бути вже удвічі більшою ОП (знову ж таки, якщо дозволяє “Список 1”).

Ці вершини фіксуються у тому ж шарі, що і a_{kj} (стовпець 7 табл. 3) і вилучаються зі “Списку 1” всіх a_{kj} . Вершину a_{kj} вилучають з підмножини A_k . Перехід до п. 4.

Так. “Список 1” порожній. Вершину a_{kj} вилучають з підмножини A_k . Перехід до п. 4.

7. Перевіряють, чи залишилися ще вершини в “Списку 1” (у підмножині A_{N-k})?

Так. Відновлюють безліч A_k . (Це означає, що весь процес повторюється, тобто знову з підмножини A_k вибиратимуться вершини a_{kj} , починаючи з першої). Перехід до п. 4.

Ні. Перехід до п. 8.

8. Кінець алгоритму.

У результаті виконання запропонованого алгоритму граф G згідно з рис. 1 виявляється перетвореним до виду, зображеного на рис. 5. Для заданої ширини l “вікна перегляду” заповнюють стовпці 2 і 3 таблиці зв'язності вершин графу; в табл. 3 у цих стовпцях наведено номери попередників і наступників кожної роботи, якщо $l = 2$. Фрагмент графу переходів процесу $X(t)$ в разі виконання КВР, зображеного на рис. 5, наведено на рис. 6.

Таблиця 3

Таблиця зв'язності вершин графу G (рис. 5)

Номер роботи a_j	Роботи-попередники $\{a_{пп}\}_j$	Роботи-наступники $\{a_{нп}\}_j$	r_j	b_j	d_j	h	Список 1
1	–	2,4	12	2	1	1	–
2	1	3,7	11	2	1	2	4
3	2	5	10	1	1	3	4,7,8
4	1	–	10	0	2	2	–
5	3	6	9	1	1	4	7,8,10,13,16
6	5	9	8	1	1	5	7,8,10,11,13,16,17
7	2	–	8	0	3	3	–
8	–	10	8	1	3	4	–
9	6	14	7	1	1	6	8,10,11,13,16,17
10	8	–	7	0	3	5	–
11	–	–	4	0	5	8	–
12	14	15	5	1	1	8	11,13,16,17,18,20,21
13	–	–	5	0	5	6	–
14	9	12	6	1	1	7	11,13,16,17
15	12	19	4	1	1	9	11,16,17,18,20,21
16	–	–	4	0	6	7	–
17	5	–	3	0	6	5	–
18	–	21	3	1	3	10	–
19	15	22	3	1	1	10	17,18,20,21
20	–	–	3	0	3	9	–
21	18	23	2	1	2	11	–
22	19	23	2	1	1	11	21
23	21,22	–	1	0	1	12	–

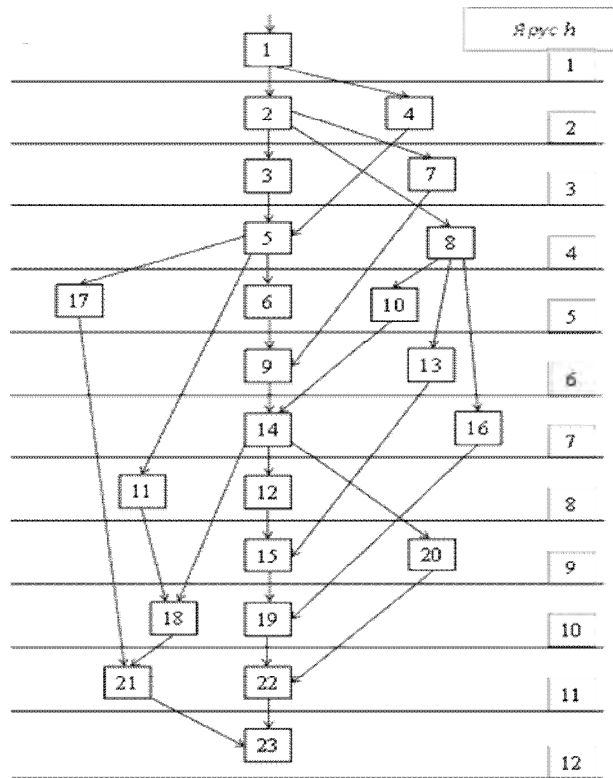


Рис. 5. Граф G заданого КВР

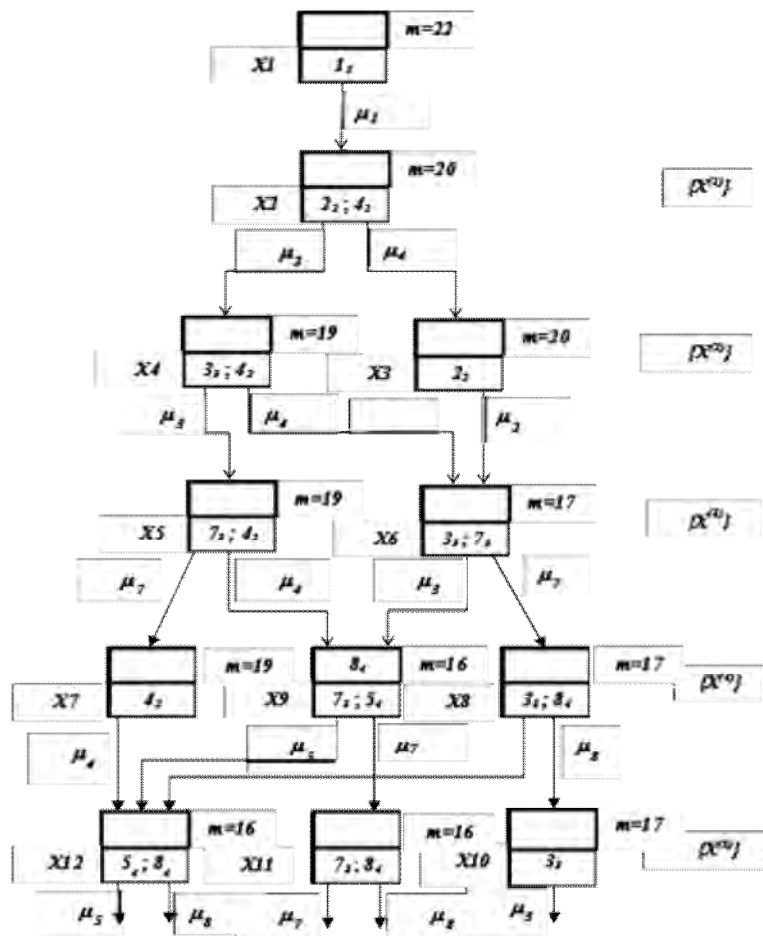


Рис. 6. Граф переходів марковського процесу

Числовий аналіз

Результати числового розрахунку середнього часу \bar{T} виконання КВР (рис. 1), якщо $l=2$, на $k=2$ обслуговувальних приладах для різних розбиттів графу G на шари наведено в табл. 4. Значення \bar{T} за прямого і поярусного стохастичного моделювання того самого КВР розраховані за тих самих значень інтенсивностей m_j обслуговування (див. стовпчики 5 у табл. 1).

З трьох розглянутих варіантів розміщення вершин графу по шарах найефективнішим, з погляду точності прогнозування, є рівномірний розподіл вершин по ярусах. У цьому випадку кількість станів процесу S та похибки методу Δ відносно $\bar{T}_{пр.ст.}$ є мінімальними: $S=63$ та $\Delta = 2,26\%$.

Таблиця 4

Результати числового розрахунку середнього часу виконання КВР

Метод моделювання	Кількість S станів ОМП	Середній час \bar{T}	Похибка $\Delta = \frac{T - T_{пр.ст.}}{T_{пр.ст.}} \cdot 100\%$	
Пряме стохастичне моделювання, $l=R=12$, Критерій $K: "r_j/b_j"$	134	1414	0	
Поярусне стохастичне моделювання	Рівномірний розподіл вершин, $l=2$	63	1446	2,26
	Розміщення вершин у ярусах з мінім. можливими номерами $l=2$	67	1508	6,65
	Розміщення вершин у ярусах з макс. можливими номерами $l=2$	63	1488	5,23

Висновки

На основі аналізу та проведених досліджень розроблено евристичний алгоритм рівномірного розподілу робіт КВР (вершин графу G) згідно з ярусами (шарами) з урахуванням кількості обслуговувальних приладів. Використання цього алгоритму дає змогу однозначно розташовувати вершини графу G згідно з ярусами (шарами) вздовж критичного шляху. Це дає можливість отримати точніші результати у разі використання методу поярусного стохастичного моделювання для прогнозування часу виконання комплексу взаємопов'язаних робіт на паралельних обчислювальних системах.

1. Hritankov A. S. *Matematycheskaja model harakteristik proizvoditenosti raspredelennyh vychislitelnyh system. Informatyka, upravlenie, ekonomika TRUDY MFTI.* – 2010. – Т. 2, No. 1(5), p. 110–115. 2. Ivanov N. N. *Matematycheskoe prognozirovanie nadezhnogo vypolnenija naborov zadach s simetrichnymi raspredelenijami vremeni vypolnenija. Xhurnal "Otkrytoe obrazovanie", vypusk No. 2–2.* – 2011, p. 52–55. 3. Ignatushenko V. V., Klushyn Ju.S. *Prognozirovanie vypolnenija slozhnyh programmnyh kompleksov na parallelnyh kompjuerah: prjamoe stohasticheskoe modelirovanie // Avtomatika i telemehanika – 1994. – No. 12, p. 142–157.* 4. Klushyn Ju. S. *Prognozirovanie vypolnenija slozhnyh programmnyh kompleksov na parallelnyh kompjuerah // Tez. Dokl. Vtoroj Ukrainskoj*

konferentzii po avtomaticheskomu upravleniju "Avtomatika – 95", Lvov, 1995, t. 2, p. 100.

5. Lysenko A. V. *Kratkij obzor metodov imitatzionnogo modelirovanija* / A. V. Lysenko, N. V. Gorjachev, I.D/Grab, B. K. Kemalov, N. K. Jurkov // *Sovremennye informatzionnye tehnologii*. 2011. – No. 14, pp. 171–176.

6. Borshchev A., Filippov A. *From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. The 22nd International Conference of the System Dynamics Society, July 25–29, 2004, Oxford, England*.

7. Lobuzov A. A. *Ispolzovanie statisticheskogo modelirovanija pri izuchenii system macovogo obsluzhyvanija*, zh. *Almanah mirovoj nauki*, izd. "Ar-Konsalt", No. 2–1(17), pp. 10–13, 2017.

8. Ignatuschenko V. V., Klushyn Ju. S. *Prognozirovanie vypolnenija slozhnyh programmnyh kompleksov na upravljajuschih paralelnyh kompjuerah: tochnye metody* // *Nauchnye trudy Mezhdunarodnogo simpoziuma "Avtomatizirovannye sistemy upravlenija"*, g.Tbilisi: izd. "Intelekt", 1996, p. 23–28.

9. Klushyn Ju. S. *Zmenschennja kilkosti staniv markivskogo protsesu pry vykonanni skladnyh programnyh kompleksiv na paralelnyh kompjuerah. Nakovyj visnyk Chernivetskogo universitetu. Kompjuterni systemy ta komponenty*. – 2016. – T. 7. Vyp. 2, pp. 53–62.

10. Ivutin A. N., Larkin E. V. *Prognozirovanie vremeni vypolnenija algoritma. Zhurnal. Izvestija TulGU. Tehnicheskie nauki. Vypusk No. 3*. – 2013, pp. 301–315.

11. Bocharov P. L., Prejdunov Ju.V. *Otzenka vremeni vypolnenija kompleksa robot na paralelnoj vychislitelnoj sisteme* // *Sistemnyj analiz i informatika. Sb. nauchnyh trudov, M.,: Izd-vo DN, 1991. pp. 29–41*.

12. Ignatuschenko V. V. *Organizatsyja struktur upravljajuschich mnogoprotzesornyh vychislitelnyh system*. – M.: Energoatomizdat. 1984.

13. Klushyn Ju. S. *Otsinka efektyvnosti riznyh dystyplin dyspetcheryzatsii dlja zmenschennja chasu vykonannja skladnyh programnyh kompleksiv na paralelnych obchusljuvalnyh systemah* / *Visnyk NU "Lvivska politehnika"* No. 413. *Kompjuterna inzhenerija ta informatijni tehnologii*. – Lviv: NU "LP". 2000. – P. 19–23.