

Ю. С. Клушин

Національний університет “Львівська політехніка”,  
кафедра електронних обчислювальних машин

## **ПРОГРАМНЕ РЕАЛІЗУВАННЯ МАТЕМАТИЧНИХ МОДЕЛЕЙ, МЕТОДІВ ТА АЛГОРИТМІВ ОЦІНЮВАННЯ ЧАСУ ВИКОНАННЯ СКЛАДАНИХ ПРОГРАМНИХ КОМПЛЕКСІВ У БАГАТОПРОЦЕСОРНИХ КОМП'ЮТЕРНИХ СИСТЕМАХ**

© Клушин Ю. С., 2018

Для вирішення проблеми прогнозування у повному обсязі розроблено програмний комплекс, який побудований на основі математичних моделей, методів та алгоритмів прямого стохастичного модулювання та поєрусного стохастичного модулювання, використовуваних для оцінювання часу виконання складаних програмних комплексів у багатопроцесорних комп'ютерних системах. Наведений програмний комплекс обчислює середнє значення і функцію розподілу часу виконання сукупності взаємопов'язаних завдань на однорідних ресурсах паралельної обчислювальної системи.

**Ключові слова:** паралельні обчислювальні системи, комплекс взаємопов'язаних робіт, пряме стохастичне моделювання, марківський процес, функція розподілу випадкової величини.

Y. Klushyn

National University “Lviv Polytechnic”,  
Computer Engineering Department

## **SOFTWARE IMPLEMENTATION OF MATHEMATICAL MODELS, METHODS AND ALGORITHMS FOR ESTIMATING THE TIME OF EXECUTION OF COMPLEX SOFTWARE COMPLEXES IN MULTIPROCESSOR COMPUTER SYSTEMS**

© Klushyn Y., 2018

To solve the forecasting problem, a software package has been developed in full, which is based on mathematical models, methods and algorithms of direct stochastic modeling and tiered stochastic modeling, which are used to estimate the execution time of folding software systems in multiprocessor computer systems. The given software package calculates the average value and the distribution function of the execution time of a set of interrelated tasks on homogeneous resources of a parallel computing system.

**Keywords:** parallel computing systems, complex of interconnected works, direct stochastic modeling, Markov process, function of distribution of random variable.

### **Вступ**

Сьогодні ефективність використання ЕОМ, зокрема, паралельних обчислювальних систем, оцінюють у ряді застосувань не стільки традиційними параметрами продуктивності (швидкістю виконання різних операцій, їх сумішей, типових обчислювальних процедур), скільки часом виконання конкретних завдань або їхніх наборів.

Такий підхід має принципове значення для оцінювання обчислювальних систем (ОС), що функціонують у контурах управління, де головним критерієм якості ОС стає її здатність розв'язати задачу за час, не більший за заданий "директивний" час. Дослідження ефективності паралельних ОС для відповідальних застосувань такого роду, ґрунтується на аналізі структури взаємозв'язків паралельно-послідовних завдань (фрагментів) заданого набору [1–3] і параметрів його паралелізму [4–8].

У зв'язку з використанням паралельних ОС у системах управління реального часу, очевидної актуальності набуває проблема апріорного оцінювання "придатності" таких ВС для розв'язання конкретного набору задач, що задається користувачем, за необхідний час.

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

Формально під прогнозуванням часу виконання конкретного комплексу робіт розуміють стохастичне оцінювання часу  $T$  його реалізації (середнє значення, дисперсія, функція розподілу  $T$ ) на паралельній ОС, а також визначення ймовірності виконання комплексу за час, не більший за заданий "директивний" час  $T_{\max}$  [2, 9, 10]

Важливо зазначити, що час виконання кожного програмного модуля (роботи) і КВР загалом розглядається тут як випадкова величина – 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 та ін. Тому використання відомих точних методів оцінювання  $T$ , наприклад, методів теорії розкладів, для вирішення проблеми прогнозування у зазначеній постановці є неприйнятним.

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 вирішити актуальне завдання достовірного аналітичного оцінювання часу виконання кожного конкретного КВР на ОС апріорно – до вибору структури і конфігурації паралельної ОС або до детального розроблення програм КВР.

### **1. Аналіз останніх досліджень та публікацій**

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 виміром значень показників працюючої системи. Завантаження ОС або її пристроїв може бути створено реальними програмами, типовими для заданої області застосування, або за допомогою спеціальних трас, які повинні якомога точно відобразити особливості реального завантаження ОС [4].

Імітаційне (статистичне) моделювання зазвичай використовують для порівняльного аналізу альтернативних архітектурних і структурних рішень при проектуванні різних вузлів ОС, а також для оцінювання точності математичного моделювання. Питання створення і застосування методів імітаційного моделювання доволі добре освітлено 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 в загальному випадку – трудомісткий і складний процес. Витрати на створення імітаційних моделей і проведення моделювання, навіть із використанням спеціалізованих мов, доволі значні.

Для дослідження ефективного функціонування і поведінки паралельних ОС на етапі їх проектування або вибору їх структури і конфігурації для передбачуваної області застосування найчастіше використовують математичне моделювання. Розрізняють детерміновані й стохастичні (ймовірнісні) моделі. Детерміновані моделі застосовують для оцінювання величини відносно 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 марківських 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 із багатьма потоками команд і даних (БКБД) – легко інтерпретуються як системи масового обслуговування завдяки таким факторам:

- паралельні функціональні блоки ОС можна трактувати як задану кількість паралельних обслуговуючих пристроїв (процесорів, пристроїв введення-виведення та ін.);
- звернення пристроїв один до одного відбувається асинхронно, у випадкові моменти часу, при цьому на входах обслуговуючих приладів формуються черги з випадковою кількістю заявок.

Існує програмна реалізація відомого підходу до проблеми прогнозування [4–6], ґрунтованого на використанні взаємопов'язаних математичних моделей:

- графової моделі заданого комплексу взаємопов'язаних робіт (КВР), що характеризується спеціальними параметрами паралелізму;
- базової моделі функціонування однорідної паралельної ОС на КВР, що задаються користувачем.

Негативні особливості цього підходу до прогнозування часу виконання КВР, що ґрунтуються на використанні параметрів паралелізму КВР, полягають у такому:

1. Надмірна формалізація і узагальненість моделі КВР, яка описується тільки числом робіт  $N$  і параметрами паралелізму  $a_i$ , "приховуючи" конкретну структуру заданого КВР.

2. Одним і тим самим значенням  $N$  та  $a_i$  (а отже, тому самому прогнозу часу виконання  $T$ ) може відповідати цілий спектр конкретних структур КВР. У зв'язку з цим точність прогнозування  $T$ , порівняно із імітаційним моделюванням реалізації цих самих КВР, змінюється в доволі широких межах. Введення ж додаткових параметрів у графову модель КВР (зокрема, параметрів критичних шляхів у графі  $G$  та його підграфів) призводить до різкого ускладнення математичної моделі загалом і до збільшення її розмірності за "неадекватного" збільшення точності прогнозу.

3. Як вже згадувалося, інтенсивність обслуговування усереднюється та приймається однаковою для всіх робіт КВР. У реальних же КВР значення інтенсивності обслуговування можуть відрізнятися доволі істотно (зокрема, на порядки) як одне від іншого, так і від своїх усереднених значень.

4. Визначення параметрів паралелізму для деяких графів КВР може виявитися неоднозначним.

## 2. Постановка задачі

Метою статті є методика побудови програмного комплексу на основі існуючих математичних моделей, методів та алгоритмів **прямого стохастичного моделювання та поярусного стохастичного моделювання**, які використовують для прогнозування часу виконання конкретних складних завдань (що задаються сукупністю взаємопов'язаних робіт – КВР) на паралельних обчислювальних системах [10–12, 17, 19]. Наведений у статті програмний комплекс вирішує проблему прогнозування в повному обсязі, тобто розраховують середнє значення часу виконання КВР  $T$  на однорідних ресурсах паралельної обчислювальної системи, а також вираховують функцію розподілу випадкової величини  $T$  – часу виконання сукупності взаємопов'язаних робіт паралельної обчислювальної системи.

## 3. Пряме стохастичне моделювання виконання КВР

Одним з таких методів, який якнайточніше вирішує проблему прогнозування часу виконання КВР, є метод прямого стохастичного моделювання [10–12].

Суть методу полягає у такому. 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 зазначено номери робіт (вершин), що є попередниками і наступниками цієї роботи. Процес виконання КВР представлений математичною моделлю (рис. 2) у вигляді однофазної СМО з  $k \geq 2$

однотипних ОП із буфером Б для готових до виконання робіт (поточний фронт  $F$ ); останні надходять з пулу П, що містить у початковому стані  $N$  робіт. Модель функціонує безперервно. Випадковий час  $t_j$  обслуговування будь-якої роботи  $a_j$  вважатимемо розподіленим 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  $M[t_j]$ .

У початковий момент часу в систему (буфер Б і обслуговуючі прилади ОП) надходить з пулу П одна заявка – “початкова” робота  $a_1$ , яку негайно обслуговують на одному з ОП. Після завершення обслуговування  $a_1$  (в 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 готові до виконання (тобто виконані всі їхні попередники); номери цих робіт однозначно встановлюють за таблицею зв'язності вершин графу  $G$ .

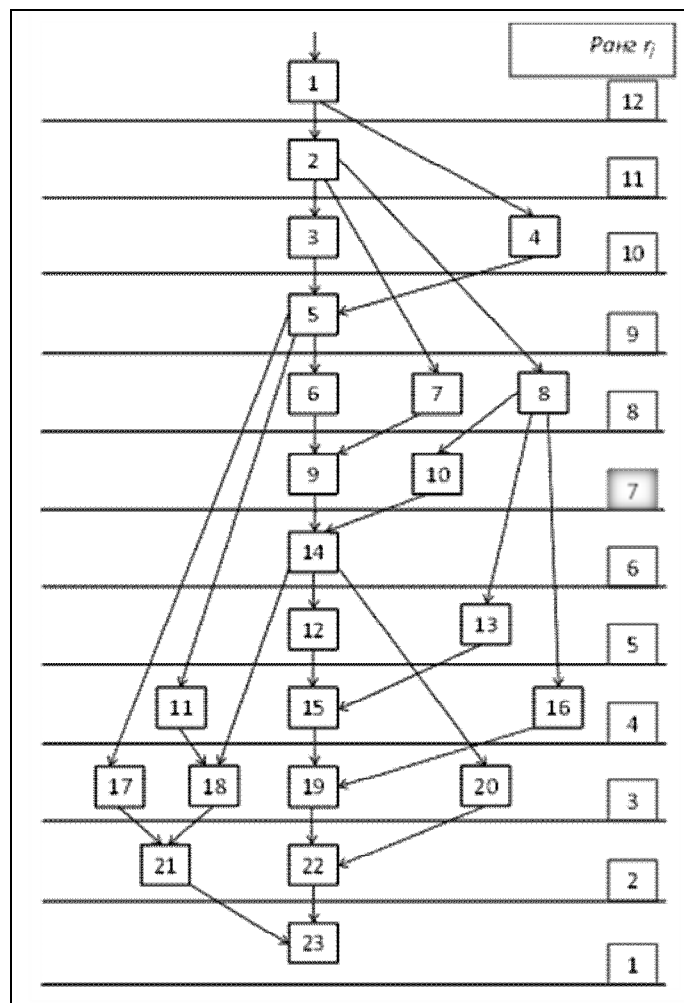


Рис. 1. Граф  $G$  заданого КВР

З буфера Б заявки вибирають випадково або за деякої іншої заданої дисципліни на обслуговування в незайнятій ОП. СМО функціонує доти, доки в пулі П і в системі не вичерпаються заявки.

У станах системи враховують: які саме роботи знаходяться в системі в поточний момент часу і які саме роботи ініціюються після завершення виконання конкретної роботи; тобто за прямого стохастичного моделювання виконання КВР відстежується конкретна структура кожного заданого КВР. Будемо вважати, що задано деяку дисципліну (критерій)  $K$  [14] для однозначного вибору робіт із буфера Б на обслуговування в ОП, наприклад, вибір роботи з найменшим номером.

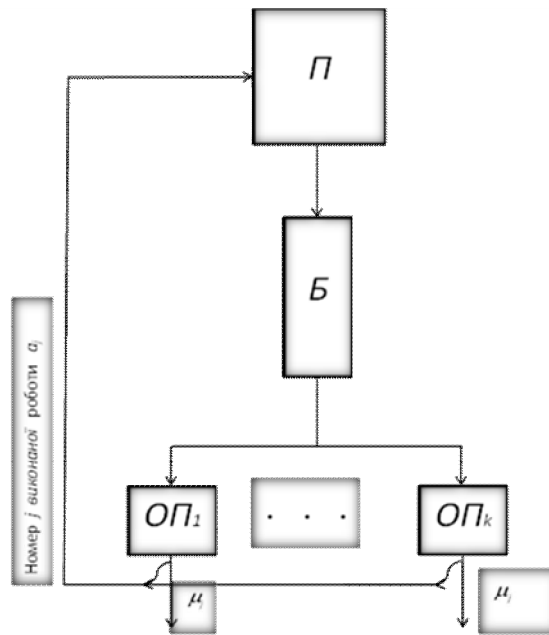


Рис. 2. Математична модель

Нехай  $T$  – час життя системи (час виконання КВР з  $N$  робіт). Функціонування цієї СМО можна описати марківським процесом, що обривається [15, 16], (ОМП),  $X(t), t \in [0, T)$ , над наступним безліччю станів:

Початковим станом процесу  $X(t)$  є стан  $X(0) = (N-1; 0, \dots, 0; 1, 0, \dots, 0)$ , за якого в пулі знаходяться  $N-1$  робіт, буфер порожній, а на обслуговуванні в ОП знаходиться робота з номером  $1$  (тобто робота  $a_1$ ).

Процес обривається, тобто переходить у поглинальний стан  $X(T) = (0; 0; 0)$  після завершення обслуговування на ОП "кінцевої" роботи  $a_N$ , що свідчить про виконання всіх  $N$  робіт заданого КВР.

Для приведеної моделі зауважимо, що одночасно в системі (Б, ОП) можуть перебувати тільки ті роботи КВР, які є паралельними, тобто можуть виконуватися одночасно одна щодо іншої.

Неважко переконатися, що будь-які дві роботи  $a_j$  та  $a_i$  є паралельними, якщо в графі  $G$  немає жодного шляху (від "початкової" вершини  $a_1$  до "кінцевої" вершини  $a_N$ ), що містить обидві вершини  $a_j$  та  $a_i$ . Очевидно, що чим більше можливих взаємно паралельних робіт визначено для цього КВР, тим більша кількість станів може мати відповідний ОМП. Тому верхню оцінку  $S^*$  числа  $S$  станів процесу  $X(t)$ , що моделює виконання будь-якого КВР з  $N$  робіт, можна визначити з "максимально паралельного" КВР, в якому всі роботи, крім "початкової" і "кінцевої" (тобто  $N-2$  робіт), є паралельними одна щодо іншої. Можна показати, що для цього випадку

$$S^* = 2 + \sum_{l=1}^k C_{N-2}^l, \quad (1)$$

де  $k$  – кількість обслуговуючих приладів;  $k = \overline{1, N-2}$ ;  $C_{N-2}^l = \frac{(N-2)!}{(N-2-l)!l!}$

Для кожного заданого КВР кількість  $S$  станів ОМП  $X(t)$  істотно залежить від конкретної структури КВР, і тому в загальному випадку підрахунок числа  $S$  аналітичним шляхом не є можливим. Однак це число може бути підраховано в результаті виконання розробленого уніфікованого алгоритму побудови безлічі  $X$  станів і матриці  $Q$  інтенсивностей переходів процесу  $X(t)$  [10–13].

Таблиця зв'язності вершин графу  $G$  (рис. 1)

Номер роботи $a_j$	Роботи - попередники $\{a_{пп}\}_j$	Роботи - наступники $\{a_{нп}\}_j$	$M[t_j], \tau$	$\mu_j$	Ранг $r_j$	Зв'язність $b_j$
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

#### 4. Блок – схема програмного комплексу

Програмний комплекс складається з п'яти модулів:

- модуль завдання структури графу;
- модуль визначення станів ОМП;
- модуль перетворення матриці  $Q$  до блочно-трикутного вигляду;
- модуль розрахунку середнього значення часу;
- модуль розрахунку функції розподілу.

На рис. 3 показано блок – схему зв'язків усіх наведених модулів.

##### 4.1. Модуль завдання структури графу КВР

Цей модуль слугує для введення в програму початкових даних і складається з двох підмодулів –  $A$  і  $B$ .

Підмодуль  $A$  слугує для введення початкових даних за використання алгоритму, описаного у розділі 4, а також у роботах [10–13]. Як вихідні дані використано:



Рис. 3. Блок-схема програмного комплексу

- таблиця зв'язності вершин графу, яка описує граф  $G$  заданого КВР (табл. 1, стовпчики 1–3);
- параметри, які характеризують заданий КВР (див. нижче);
- параметр паралельної ОС – кількість  $k$  обслуговуючих приладів (процесорів ОС);
- параметр  $cr$ , який задає критерій диспетчеризації.

Параметрами, які характеризують заданий КВР, є:

- $N$  – загальна кількість усіх робіт заданого КВР (або початкова кількість заявок у пулі П);
- $\mu_j$  – інтенсивність обслуговування кожної роботи (заявки) (табл. 1, стовпець 5);
- $b_j$  – ступінь зв'язності кожної роботи заданого КВР, що визначає кількість наступників

(підмножина  $\{a_{HP}\}_j$ ) роботи  $a_j$  (табл. 1, стовпець 3);

- $r_j$  – ранг роботи  $a_j$  у заданому КВР.

Параметр  $cr$ , який задає критерій диспетчеризації, може набувати таких значень:

- 0 – диспетчеризація за критерієм рангу: “ $r_j$ ” (табл. 1, стовпець б);
- 1 – диспетчеризація за критерієм рангу і зв'язності вершин “ $r_j / b_j$ ”;
- 2 – диспетчеризація за критерієм “вибір роботи із найменшим номером”;
- 3 – диспетчеризація за критерієм “вибір роботи із найбільшим номером”.

*Підмодуль В* слугує для завдання початкових даних при використанні алгоритмів поярусного стохастичного моделювання [19]. У цьому підмодулі використано також алгоритм рівномірного розподілу робіт заданого КВР за ярусами [17]. Як вихідні дані використовують таблицю зв'язності [17] і зазначені вище параметри ( $N, \mu_j, b_j, r_j$ ).

#### 4.2. Модуль визначення станів ОМП при виконанні конкретного КВР

Модуль реалізує алгоритми, які викладені у розділі 2, а також у роботах [10–13, 17, 19]. Задається початковий стан системи.

*Стан\_системи*

$$\begin{aligned}
 \text{Роботи\_пула} &= N-1 \\
 \text{Роботи\_буфера} &= 0 \\
 \text{Роботи\_ОП} &= 1 \text{ (тобто робота } a_1) \\
 \text{Кількість\_станів} &= 1 \\
 \text{Поточний\_стан} &= 1
 \end{aligned}$$

У модулі використано три робочі масиви:

- масив  $Q(s, s)$ , який визначає матрицю  $Q$  інтенсивності переходів розглянутого ОМП;
- масиви  $X1()$ ,  $X2()$ , елементами яких є стан системи  $X_r(m, iw, jn)$ .

Слід зазначити, що при прямому стохастичному моделюванні розмірність масиву  $Q(s,s)$  в принципі можна оцінити за формулою (1). Це доцільно для КВР із невеликою кількістю робіт, коли кількість станів ОМП свідомо незначна. За великих розмірностей моделі різко збільшується обсяг обчислень, необхідних для обробки матриці  $Q$ , а отже, і витрати пам'яті комп'ютера. Так, для графу, наведеного на рис. 1, кількість станів відповідно до формули 1 дорівнює 233, реально ж  $S = 134$  (табл. 1). Тому для зменшення обсягу використовуваної комп'ютерної пам'яті бажано визначити реальну розмірність масиву  $Q(s,s)$ , що і є одним з результатів реалізації алгоритмів визначення станів ОМП [10–12, 17, 19] у цьому програмному модулі. Додатковою можливістю для економії обсягу комп'ютерної пам'яті при реалізації запропонованих методів є спеціальна нумерація і впорядкування станів у процесі побудови матриці  $Q(s,s)$  [10–12].

#### **4.3. Модуль перетворення матриці інтенсивностей переходів до блочно-трикутного вигляду**

Цей програмний модуль не лише перетворює матрицю  $Q$  інтенсивностей переходів до блочно-трикутного вигляду, але і формує цю матрицю в стислій формі, що спрощує і прискорює виконання алгоритмів отримання чисельних характеристик досліджуваної системи.

Для приведення матриці до трикутного вигляду необхідно і достатньо впорядкувати елементи вектора станів  $X$ . Нумерацію і впорядкування станів, як і в [18], створюємо за спаданням або незростанням значень параметра  $m$  (поточна кількість робіт у пулі  $\Pi$ ) так, щоб значення  $m$  для кожного стану  $X_p \in X$  виявлялося б не більшим, ніж у будь-якому з попередніх (з меншим номером) станів. Стани з однаковими значеннями  $m$  упорядковують за спаданням або незростанням значень  $c = w + n$  (кількістю робіт у системі); якщо ж два або більше станів характеризуються однаковими значеннями  $m$  та  $c$ , то їх нумерують довільно.

Для зменшення обсягу використовуваної пам'яті і прискорення розрахунку використано спеціальне подання нижньої трикутної матриці  $Q$  розмірності  $n * n$  у вигляді двох масивів:

- масиву  $wq(m,p)$  значень нульових елементів у рядку матриці  $Q$ ;
- масиву  $iq(m,n)$  номерів ненульових елементів у рядку матриці  $Q$ .

Оскільки в матриці  $Q$  усі діагональні елементи відмінні від нуля, то в масиві  $iq(m,n)$  перший стовпець використано для позначення типу рядка, в якому розташовано описані в ній ненульові елементи. У модулі також використано масив  $num(p)$  кількості ненульових елементів у рядках різних типів і масив  $vec(n)$  вектора ймовірностей початкових станів системи розмірності  $s$ .

#### **4.4. Модуль розрахунку середнього значення часу виконання КВР**

За допомогою цього модуля розраховують середнє значення часу виконання КВР  $T$  на однорідних ресурсах паралельної ОС. При цьому у модулі використовують один робочий масив  $T(s,4)$ , в якому запам'ятовується поточне значення  $T$ .

#### **4.5. Модуль розрахунку функції розподілу**

За допомогою цього модуля вираховують функцію розподілу випадкової величини  $T$  – часу виконання комплексу взаємопов'язаних робіт паралельної ОС. Алгоритм знаходження функції розподілу випадкової величини описано в роботі [18].

### **Висновки**

На основі аналізу математичних моделей, методів та алгоритмів прямого стохастичного модулювання та поярусного стохастичного модулювання, а також проведених досліджень, створено програмний комплекс для вирішення проблеми прогнозування часу виконання складних розгалужених завдань на паралельних обчислювальних системах у повному обсязі. Наведений програмний комплекс обчислює середнє значення і функцію розподілу часу виконання сукупності взаємопов'язаних завдань на однорідних ресурсах паралельної обчислювальної системи і складається з п'яти програмних модулів, які пов'язані між собою. Взаємозв'язок модулів наведено на рис. 3.



Його характеристики на тестовому прикладі:

- загальний обсяг використовуваної пам'яті комп'ютера для обробки матриці інтенсивностей переходів, що відповідає виконанню сукупності взаємопов'язаних робіт з  $N < 100$  роботами – близько 260 кбайт;
- час побудови матриці інтенсивностей переходів на комп'ютері середньої потужності становить – близько 7 хв;
- час розрахунку моделі загалом – близько 16 хв.

1. Chu W. W., Leung K.K. *Module replication and assignment for real-time distributed processing system.* // "Proc IEEE". 1987. 75. N5. pp. 547–562. 2. Khritankov A. S. *Mathematical model of performance characteristics of distributed computing systems. Computer science, management, economics. WORKS OF MIPT.* – 2010. – Volume 2, No. 1 (5), p. 110–115. 3. Ivutin A. N., Larkin E. V. *Prediction of the execution time of the algorithm. Magazine. News of TSU. Technical science. Issue number 3/2013* C. 301–315. 4. Bocharov P. L., Ignatushchenko V. V. *Mathematical models and methods for evaluating the effectiveness of parallel computing systems on complexes of interrelated works* // Tez. report international conf, "High-Performance Computing Systems in Management and Scientific Research," Alma-Ata, 1991, p. 6. 5. Margalitashvili, A. L., Ambartsumian, A. L., Teplyakov, A. V., Preidunov, Yu. V., *Graph models of complexes of interrelated works, M \**, 1990, – Dep. in VINITI 31,01,90, No. 587–B90. 6. Margalitashvili A. L., *Investigation of the effectiveness of the functioning of parallel computing resources on given complexes of interrelated works, Abstract of Cand. dis. M. : In-t prbblem management, 1990.* 7. Bocharov P. L., Preidunov Yu. V., *Estimation of the execution time of a complex of works on a parallel computational system* // *System analysis and computer science. Sat scientific papers, M.: Publishing house DN, 1991.* C 29–41. 8. Ingatushchenko V. V. *Organization of structures for controlling multiprocessor computing systems. Moscow: Energoatomizdat, 1984.* 9. Ivanov N.N. *Mathematical prediction of reliable execution of sets of tasks with symmetric runtime distributions. Journal of Open Education, Issue No. 2–2 / 2011,* p. 52–55. 10. Ignatushchenko V. V., Klushin Y. S. *Prediction of the implementation of complex software systems on parallel computers: direct stochastic modeling* // *Automation and Remote Control. 1994. N12,* p. 142–157. 11. Klushin, Y. S. *Prediction of the implementation of complex software systems on parallel computers* // *Proc. Report Second Ukrainian Conference on Automatic Control "Automation – 95", Lviv, 1995, vol. 2,* p. 100. 12. Ignatushchenko V. V., Klushin Yu. S. *Forecasting the implementation of complex software systems on control parallel computers: exact methods* // *Scientific works of the International Symposium "Automated Control Systems", Tbilisi: ed. Intellect, 1996,* p. 23–28. 13. N. N. Ivanov, V. V. Ignatushchenko, A. Y. Mikhailov, *Static prediction of the execution time of complexes of interrelated works in multiprocessor computing systems, Avtomat. and Telemekh., 2005, issue 6,* 89–103. 14. Klushin Y. S. *Evaluation of the effectiveness of various dispatching disciplines for reducing the time to perform complex software systems on parallel computing systems / Bulletin of National University "Lviv Polytechnic" No413. Computer engineering and information technology. – Lviv: NU "LP", 2000. – p. 19–23.* 15. Gross, D., Miller, D., *Transition Markov processes* // *Operations Research. 1984. Vol. 32. No 4. P. 334–361.* 16. Reibman A. L., Trivedi K. S. *Numerical transient analysis of Markov models* // *Computers and Operations Research. 1988. V. 15. No. 1. P. 19–36.* 17. Klushin, Y. S. *Improving the accuracy of estimating the execution time of folding software systems in multiprocessor computer systems for belt stochastic modeling. Bulletin of NU "Lviv Polytechnic" No881. Computer systems and networks. – Lviv: NU "LP", 2017.* 18. Preidunov Y. V. *Development of mathematical models and methods for predicting the implementation of complex software systems on parallel computing systems. Cand. course work. M. : Inst. Of Problems of Management RAS, 1992.* 19. Klushin Y. S. *reducing the number of states of the Markov process when executing complex software systems on parallel computers. Scientific Bulletin of Chernivtsi University. Computer systems and components. 2016. T. 7. Vol. 2, pp. 53–62.*