

ОЦІНЮВАННЯ НАДІЙНОСТІ ПАРАЛЕЛЬНИХ ОБЧИСЛЮВАЛЬНИХ СИСТЕМ ПІД ЧАС ВИКОНАННЯ ЗАДАНОГО КОМПЛЕКСУ ВЗАЄМОПОВ'ЯЗАНИХ РОБІТ

Ю. С. Клушин

Національний університет “Львівська політехніка”,
кафедра електронних обчислювальних машин

© Клушин Ю. С., 2019

Під час проектування паралельної обчислювальної системи для виконання заданого комплексу взаємопов'язаних робіт важливою характеристикою є її надійність. Але крім основних показників надійності обчислювальної системи, забезпечених технічними засобами, існує ще показник надійності системи, який оцінює придатність паралельної обчислювальної системи до вирішення конкретних задач за визначений час. Це оцінювання безпосередньо пов'язане з ефективністю використання обчислювальних систем.

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

Вступ

Однією з ключових характеристик комп'ютерних систем є надійність, завдяки чому можливий варіант зі завданням оцінювання показників надійності наявної системи та завданням синтезу системи зі заданими показниками надійності. Надійність технічної системи, як правило, визначають її стійкістю до відмов, а основним засобом забезпечення відмовостійкості є резервування [1–3], тобто введення надмірності.

Поняття надійності охоплює не тільки апаратні засоби, а й програмне забезпечення, яке використовують, зокрема, для аналізу продуктивності систем і управління конфігураціями. Головною метою підвищення надійності систем є цілісність зберігання в них даних. Одиницею вимірювання надійності є середній час напрацювання на відмову, інакше – середній час безвідмовної роботи.

Відмовостійкість – це здатність обчислювальної системи продовжувати дії, задані програмою, після виникнення несправності. Введення відмовостійкості потребує надлишкового апаратного і програмного забезпечення. Напрями, пов'язані із запобіганням відмов у системи, є основними для забезпечення надійності.

Отже, основними показниками для обчислювальної системи, що характеризують надійність і мають свою розмірність, є середній час напрацювання на відмову, середній термін служби, середній час відновлення, середній термін зберігання.

Для паралельних обчислювальних систем концепції паралельності й відмовостійкості пов'язані між собою, оскільки в обох випадках необхідні додаткові функціональні компоненти. Тому на паралельних обчислювальних системах досягають і найвищу продуктивність, і, здебільшого, дуже високу надійність. Наявні ресурси надмірності в паралельних системах можна

гнучко використовувати і для підвищення продуктивності, і для підвищення надійності. Структура багатопроцесорних і багатомашинних систем пристосована до автоматичної реконфігурації і забезпечує можливість продовження роботи системи після виникнення відмов.

Але крім базових показників надійності обчислювальної системи, забезпечених технічними засобами, існує ще показник надійності системи, який оцінює придатність паралельної обчислювальної системи вирішувати конкретні задачі за визначений час. Це оцінювання безпосередньо пов'язане з ефективністю використання обчислювальних систем.

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

Такий підхід має принципове значення для оцінювання обчислювальних систем (ОС), що функціонують у контурах управління, де головним критерієм якості ОС стає її здатність вирішити задачу за час, не більший від заданого “директивного” часу. Дослідження ефективності паралельних ОС для таких відповідальних застосувань ґрунтується на аналізі структури взаємозв'язків паралельно-последовних завдань (фрагментів) заданого набору [4–6] і параметрів його паралелізму [7–10].

У зв'язку з використанням паралельних ОС у системах управління реального часу очевидну актуальність набуває проблема апріорної оцінки “придатності” таких ОС для вирішення конкретного набору задач від користувачів за необхідний час.

Стосовно паралельних ОС цю проблему називають *прогнозуванням* часу виконання складних програмних комплексів; останніх зазвичай задають графовими моделями і розглядають як *комплекси взаємопов'язаних робіт* (КВР) – задач і / або їх паралельно-последовних фрагментів (підзадач, процесів, програмних модулів).

Рішення математичної задачі прогнозування часу T виконання КВР спрямовано на отримання оцінки “придатності” паралельної ОС на виконання конкретної задачі.

Таке трактування проблеми прогнозування має й іншу технічну інтерпретацію, пов'язану з надійністю обчислювальної системи: якщо розглядати невиконання конкретного КВР за заданий час T як відмову обчислювальної системи, то яка ймовірність появи такої відмови за час T або яким має бути час T , щоб із заздалегідь заданою ймовірністю відмови не було. Якщо використовувати таку технічну інтерпретацію, то відмова обчислювальної системи зводиться до неправильного проектування системи або до неправильного вибору конфігурації системи (зокрема, числа процесорів).

Аналіз останніх досліджень та публікацій

Аналіз надійності складних систем, що складаються з великої кількості елементів і мають конфігурацію, яка не зводиться до послідовно-паралельного вигляду, – це непросте завдання не тільки з огляду на неможливість створення аналітичної моделі, але і потребує великих обсягів обчислень у разі використання імітаційного програмного моделювання.

Для вирішення завдання аналізу надійності систем із застосуванням програмного моделювання зазвичай використовують два методи:

- метод статистичних випробувань;
- моделювання поведінки системи за інтервалами часу.

Перший метод ґрунтується на статистичних алгоритмах розрахунку надійності обчислювальних систем. Цей метод застосовують під час дослідження надійності складних відновлюваних обчислювальних систем. Сутність алгоритму, в основу якого покладено метод статистичних випробувань Монте-Карло, полягає в тому, що багато разів відтворюється деяка формалізована схема, що є формальним математичним описом процесу функціонування реальної системи і математичною моделлю. Результуючу ймовірність подають у вигляді математичного очікування функції випадкових величин і обчислюють наближено як середнє значення на основі доволі великої кількості випробувань.

Метод моделювання поведінки системи за інтервалами часу найширше застосовують серед методів аналізу надійності. Він найуніверсальніший і його можна використовувати під час аналізу практично будь-якої системи. Завдяки цьому опис програмної моделі системи доволі простий і однаковий для різних систем. Метод оснований на аналізі стану системи в будь-який момент часу дослідження, який визначають вибраним значенням інтервалу й обчислення ймовірності знаходження системи в цьому стані.

Крім того, існують методики забезпечення відмовостійкості багатоагентних систем (МАС) Brokored MAS [11], DARX [12], Meta-Agent [13] та робота [14], засновані на надмірності агентів або надмірності агентів і деяких завдань. Такі методики підтверджують збільшення рівня надійності багатоагентних систем, тобто зниження кількості відмов, тільки експериментально.

За всіма цими методиками не можна оцінити придатність обчислювальної системи до вирішення конкретних задач за визначений час.

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

Експериментальне дослідження ефективності ОС здійснюють безпосереднім виміром значень показників працюючої системи. Завантаження ОС або її пристроїв можна створити реальними програмами, типовими для заданої ділянки застосування, або за допомогою спеціальних трас, які повинні, якщо змога, повно відображати особливості реального завантаження ОС [7].

Імітаційне (статистичне) моделювання зазвичай використовують 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 висвітлені в літературі (див., наприклад, [8–10]). Перевагою імітаційного моделювання є можливість аналізу роботи ОС практично з будь-яким ступенем деталізації. Однак дослідження ОС методами імітаційного моделювання загалом трудомісткий і складний процес. Витрати на створення імітаційних моделей і моделювання, навіть у разі використання спеціалізованих мов, доволі великі.

Для дослідження ефективного функціонування і поведінки паралельних ОС на етапі їх проектування або вибору їх структури і конфігурації для передбачуваної ділянки застосування найчастіше використовують математичне моделювання. Розрізняють детерміновані й стохастичні (ймовірні) моделі. Детерміновані моделі застосовують під час оцінювання величини щодо 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 багатопроекторні системи з багатьма потоками команд і даних (БКБД) – легко інтерпретуються як системи масового обслуговування через такі фактори:

- паралельні функціональні блоки ОС можна трактувати як задану кількість паралельних обслуговуваних пристроїв (процесорів, пристроїв введення-виведення та ін.);
- звернення пристроїв один до одного відбуваються асинхронно, у випадкові моменти часу, завдяки цьому на входах обслуговуваних приладів формуються черги із випадковим числом заявок.

Постановка завдання

Мета статті – підвищити надійність роботи паралельної обчислювальної системи під час вирішення поставленого завдання. Запропоновано методику усунення відмов, які виникають за неспроможності паралельної обчислювальної системи виконати завдання за визначений час. У статті приведено алгоритм поетапного проектування високонадійної обчислювальної системи для виконання конкретного комплексу взаємопов'язаних робіт.

Алгоритм перевірки надійності обчислювальної системи

Під час проектування паралельної обчислювальної системи з огляду на її надійність виконання заданого комплексу взаємопов'язаних робіт (КВР) за “директивний” час потрібно використовувати такий узагальнювальний алгоритм перевірки її надійності:

1. Вибираємо конфігурацію паралельної обчислюваної системи.

Тут можуть бути різні системи, наприклад, багатомашинні паралельні системи, мульти-комп'ютерні паралельні системи або багатопроцесорні системи кластерного типу.

2. Визначаємо початкову кількість обслуговуваних пристроїв (ОП) та їхню продуктивність.

Під обслуговуваними пристроями розуміють обчислювальні компоненти системи. Наприклад, це може бути окремий процесор, окремий комп'ютер або кластер.

3. Розраховуємо ймовірність відмови обчислювальної системи у разі виконання заданого КВР за “директивний” час.

4. Перевіряємо ймовірність відмови. Якщо ймовірність відмови відповідає заданій, то переходимо до п. 5. Якщо вона більша за задану, то тоді перевіряємо зміну цій ймовірності зі збільшенням кількості ОП. Якщо ймовірність відмови змінилася, то додаємо ще один ОП і повертаємося до п. 3. Якщо ні – тоді перехід до п. 1 для зміни конфігурації системи або в п. 2 для зміни ОП.

5. Перевіряємо вартість спроектованої системи. Якщо вона перевищує задану вартість, то переходимо до п. 6. Якщо ні, то кінець алгоритму.

6. Змінюємо алгоритм роботи заданого КВР, що охоплює:

– розпаралелювання обчислень на рівні команд, виразів, програмних модулів, окремо виконуваних завдань;

– етапи побудови паралельних алгоритмів і програм, вибір паралельного алгоритму;

– реалізація алгоритму у вигляді паралельної програми;

– побудова виконуваної програми для паралельної обчислювальної системи;

– вибір оптимального алгоритму для конкретної обчислювальної системи, знаходження найкращої топології обчислювальної системи для вирішення певної задачі, розпаралелювання наявного алгоритму;

– декомпозиція алгоритму на паралельно виконуваних фрагменти обчислень;

– розподіл завдань за процесорами і балансування;

– синхронізація всіх обчислювальних процесів та взаємодія між ними.

Повертаємося до п. 1.

Виконуючи цей алгоритм, зауважимо, що відповідно до закону Амдаля прискорення процесу обчислень у разі використання k процесорів обмежується величиною

$$V \leq \frac{1}{f + (1-f)}, \quad (1)$$

де f – частка послідовних обчислень у вживаному алгоритмі опрацювання даних (тобто, наприклад, за наявності всього 10 % послідовних команд у виконуваних обчисленнях, ефект використання паралелізму не може перевищувати 10-кратного прискорення опрацювання даних).

Це зауваження характеризує одну з найсерйозніших проблем у галузі паралельного програмування (алгоритмів без певної частки послідовних команд практично не існує). Однак часто частка послідовних дій характеризує неможливість паралельного вирішення завдань. Як результат, частка послідовних обчислень може бути істотно знижена у разі вибору придатніших для розпаралелювання алгоритмів.

У використанні наявних паралельних ОС із визначеною структурою для вирішення поставленої задачі зміна конфігурації ОС здебільшого неможлива, тому основним заходом для підвищення надійності є шостий пункт зазначеного вище алгоритму.

Опис математичної моделі

Одним із основних пунктів поданого вище алгоритму є визначення ймовірності відмови обчислювальної системи під час виконання заданого КВР за “директивний” час. Для розрахунку

ймовірності відмови використовуємо метод прямого стохастичного моделювання [15–17]. Цей метод найточніше вирішує проблему надійності ОС у сенсі виконання конкретного КВР за заданий час T .

Суть методу полягає в такому.

Граф G заданого КВР (рис. 1) описано у таблиці зв'язності його вершин (табл. 1). Таблиця містить N рядків (за кількістю вершин графа G), у кожному з яких вказано номери робіт (вершин), що є попередниками і наступниками цієї роботи.

Процес виконання КВР зображено математичною моделлю (рис. 2) у вигляді однофазної СМО з $k \geq 2$ однотипних ОП із буфером B для готових до виконання робіт (поточний фронт F); останні надходять з пулу Π , що містить у початковому стані N робіт.

У початковий момент часу в систему (буфер B і обслуговчі прилади ОП) надходить із пулу Π одна заявка-“початкова” робота a_1 , яка негайно починає обслуговуватися на одному з ОП. Після завершення обслуговування a_1 (загалом – a_j) в ОП ця робота залишає систему, “передаючи” свій номер j в пул Π , з якого вибирають у буфер B тих наступників роботи a_j , які виявилися готові до виконання (тобто виконані всі їхні попередники); номери цих робіт однозначно встановлюють за таблицею зв'язності вершин графа G .

Із буфера B заявки вибирають випадковим способом або за деякої іншої заданої дисципліни на обслуговування в незайнятій ОП. СМО функціонує доти, поки в пулі Π і в системі не вичерпаються заявки.

У станах системи враховується: які саме роботи знаходяться в системі в поточний момент часу і які саме роботи ініціюються після завершення виконання конкретної роботи; тим самим у разі прямого стохастичного моделювання виконання КВР відстежується конкретна структура кожного заданого КВР.

Вважатимемо, що задана деяка дисципліна (критерій) K [18] для однозначного вибору робіт із буфера B на обслуговування в ОП, наприклад, вибір роботи з найменшим номером.

Нехай T -час життя системи (час виконання КВР з N робіт). Функціонування заданої СМО описують марківським процесом, що обривається [19, 20], (ОМП), $X(t)$, $t [0, T)$, над такою 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).

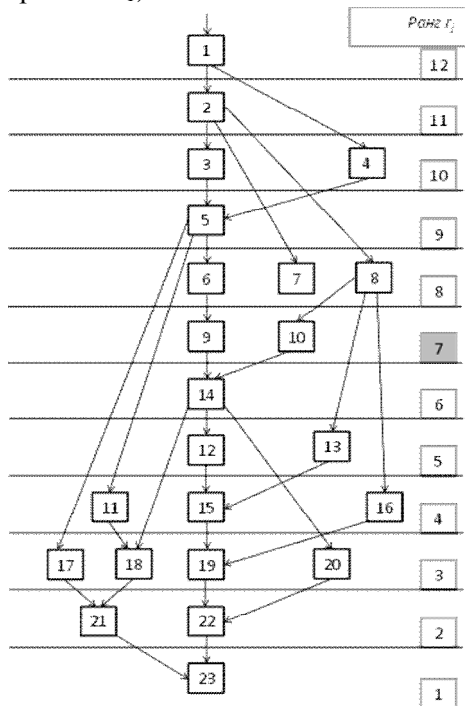


Рис. 1. Граф G заданого КВР

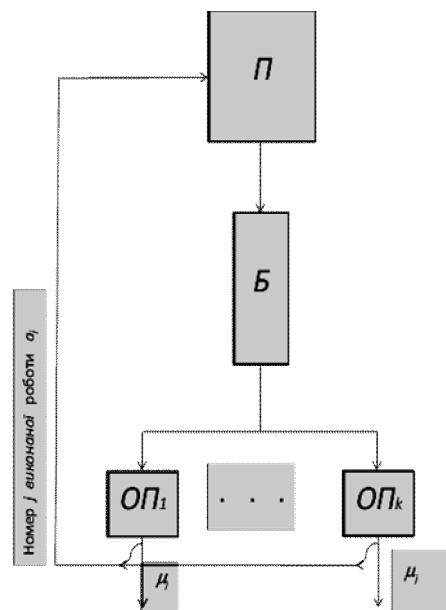


Рис. 2. Математична модель

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

Таблиця 1

Таблиця зв'язності вершин графа G (рис. 1)

Номер роботи a_j	Роботи-попередники $\{a_{пп}\}_j$	Роботи-наступники $\{a_{нп}\}_j$	$M[t_j], \square$	μ_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

Для поданої моделі зауважимо, що одночасно в системі (B , 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_i до “кінцевої” вершини a_N), що охоплює обидві вершини a_j та a_i .

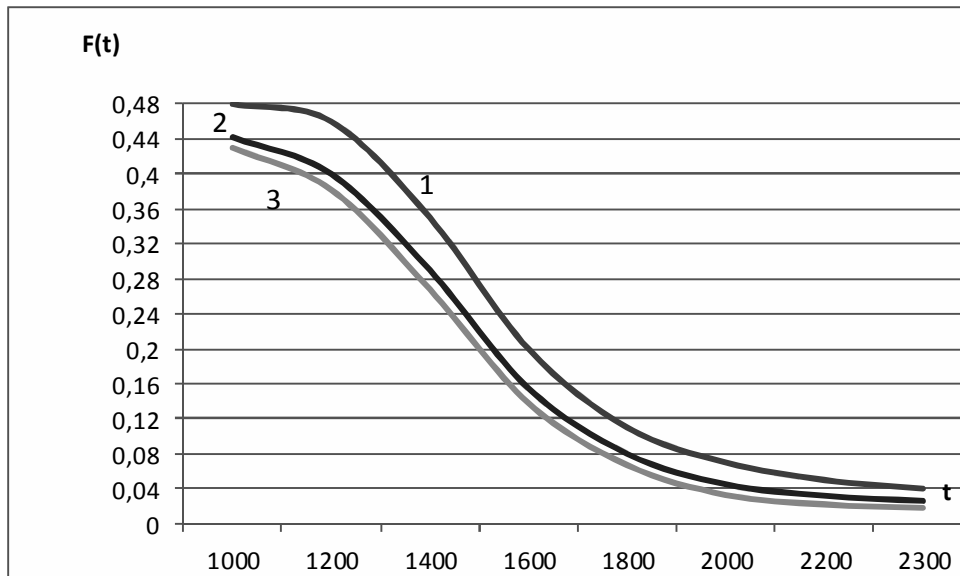
Очевидно, що чим більше можливих взаємно паралельних робіт визначено для заданого КВР, то більшу кількість станів може мати відповідний ОМП. Тому верхню оцінку S^* числа S станів процесу $X(t)$, що моделює виконання будь-якого КВР з N робіт, можна визначити з “максимально паралельного” КВР, в якому всі роботи, крім “початкової” і “кінцевої” (тобто $N-2$ робіт), є паралельними одна щодо іншої. Можна показати, що для цього випадку

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

де k – кількість обслуговуваних 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 загалом підрахунок числа S аналітичним способом не є можливим. Однак це число може бути визначено в результаті виконання розробленого уніфікованого алгоритму побудови безлічі X станів і матриці Q інтенсивностей переходів процесу $X(t)$ [21].

Рис. 3. Функція розподілу $F(t)$

Нехай $X(t)$, $t \in [0, \tau)$ – однорідний ОМП, визначений на кінцевій безлічі станів $X = \{0, 1, 2, \dots, M\}$, де τ означає час до поглинання (зупинки) процесу $X(t)$. Крім того, позначимо через $p_i(t) = P\{X(t) = i, t > \tau\}$ ймовірність стану процесу $X(t)$ у момент часу t .

Розглянемо функцію розподілу

$$F(t) = P\{\tau \geq t\} = \sum_{i=0}^M p_i(t), \quad (3)$$

яка є ймовірністю того, що за час t процес $X(t)$ не перейде в поглинаючий стан (не виродиться). Зауважимо, що зі зростанням t функція розподілу $F(t) \rightarrow 0$.

Знаходження функції розподілу дає можливість оцінити ймовірність невиконання таких комплексів за деякий заданий час на основі нестационарної математичної моделі функціонування паралельної ОС, описуваної марківським процесом, що обривається (ОМП). Цей підхід дає можливість аналітично вирішити таке практичне завдання: отримати оцінку ймовірності невиконання складних комплексів робіт на паралельній ОС за час, не більший за наперед заданий “директивний” час T . Для побудови функції розподілу $F(t)$ часу виконання заданого КВР (рис. 1) використовуємо програмний комплекс, описаний у роботах [21, 22].

На рис. 3 показано графіки функції розподілу $F(t)$ для двох ($k=2$), трьох ($k=3$) та чотирьох ($k=4$) процесорних систем.

Чисельний аналіз

Побудова функції розподілу часу невиконання конкретного КВР на паралельній обчислювальній системі дає можливість оптимального вибору кількості процесорів. Для заданого конкретного КВР (рис. 1) розглянуто три обчислювальні системи з двома, трьома та чотирма процесорами. На основі побудованих математичних моделей отримано три функції розподілу (графіки 1, 2, 3). З аналізу графіків бачимо, що ймовірність невиконання комплексу робіт за середній час (для двопроцесорної моделі $T = 1414$ тактів, для трипроцесорної $T = 1330$ і для чотирипроцесорної $T = 1280$) дорівнює приблизно 0,34; а за заданого “директивного” часу, наприклад, $T = 2300$, ця ймовірність зменшується з 0,04 (для $k=2$) та 0,026 (для $k=3$) до 0,018 для чотирипроцесорної моделі. Отже, на основі отриманих даних можна визначити ймовірність відмови обчислювальній системі за “директивного” часу виконання заданого КВР. Збільшення кількості процесорів моделі для виконання заданого КВР, як показують графіки, не дає істотного результату у зменшенні часу виконання, а значить і в зменшенні ймовірності відмови, оскільки заданий граф

має велику частку послідовних операцій – близько 50 %. Оптимальна кількість процесорів для виконання заданого КВР дорівнює 4, тому для подальшого зменшення часу виконання заданого КВР треба використовувати шостий пункт поданого вище алгоритму.

Висновок

Отже, використання методу прямого стохастичного модулювання дає можливість вирішити ще одну важливу проблему під час проектування паралельних обчислювальних систем – проблему надійності цих систем виконувати поставлені завдання за заданий “директивний” час. Завдяки цьому важлива побудова функції розподілу часу невиконання конкретного КВР на обраній математичній моделі й вибору оптимальної кількості процесорів із оптимальною продуктивністю паралельної обчислювальної системи. За неможливості досягнення директивного часу виконання заданого КВР на спроектованій або наявній паралельній обчислювальній системі необхідно змінювати алгоритм роботи чи використовувати високопродуктивніші компоненти системи. Зауважимо, що використання високопродуктивніших компонентів може призвести до значних витрат, а отже і до збільшення вартості такої обчислювальної системи.

Список літератури

1. Kuznetsov P. A. *On the issue of analysis of the effectiveness of systems with full redundancy*. *Bulletin of SibSAU*, t. 16, No. 2, p. 326–330, 2015.
2. Pullum L. L. *Software fault tolerance techniques and implementation*. Artech House, 2001. 360 p.
3. Roganov V. R., Grishko A. K., Kochegarov A. K. *Three Approaches to evaluating the Performance of active Reservation Systems*. DOI 10.21685/2307-4205-2019-2-2.
4. Chu W. W., Leung K. K. *Module replication and assignment for real-time distributed processing system // “Proc IEEE”*. 1987. 75. No. 5. pp. 547–562.
5. 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.
6. Ivutin A. N., Larkin E. V. *Prediction of the execution time of the algorithm*. *Magazine. News of TSU. Technical science*. Issue number 3/2013. С. 301–315.
7. 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*.
8. 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 prbbblem management, 1990.
9. Bocharov P. L., Preydunov 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*. С. 29–41.
10. Ingatushchenko V. V. *Organization of structures for controlling multiprocessor computing systems*. Moscow: Energoatomizdat, 1984.
11. Kumar S., Cohen P. R. *Towards a fault-tolerant multi-agent system architecture*. In: *Proceedings of the Fourth International Conference on Autonomous Agents*. ACM, 2000, pp. 459–466. DOI:10.1145/336595.337570.
12. Guessoum Z., Briot J. P., Faci N. *Towards Fault-Tolerant Massively Multiagent Systems*. In: *Massively Multi-Agent Systems I*. Springer Berlin Heidelberg, 2005. P. 55–69. (Ser. *Lecture Notes in Computer Science*; vol. 3446). DOI: 10.1007/11512073_5.
13. Serugendo G. D.M., Romanovsky A. *Designing Fault-Tolerant Mobile Systems*. In: *Scientific Engineering for Distributed Java Applications*. Springer Berlin Heidelberg, 2003. P. 185–201. (Ser. *Lecture Notes in Computer Science*; vol. 2604). DOI: 10.1007/3-540-36520-6_17.
14. Mellouli S. *A Reorganization Strategy to Build Fault-Tolerant Multi-Agent Systems*. In: *Advances in Artificial Intelligence*. Springer Berlin Heidelberg, 2007. P. 61–72. (Ser. *Lecture Notes in Computer Science*; vol. 4509). DOI: 10.1007/978-3-540-72665-4_6.
15. 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. N 12, p. 142–157.
16. 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.

17. 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.
18. 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"* No. 413. *Computer engineering and information technology*. Lviv: NU "LP", 2000. p. 19–23.
19. Gross, D., Miller, D., *Transition Markov processes* // *Operations Research*. 1984. Vol. 32. No. 4. P. 334–361.
20. Reibman A. L., Trivedi K. S. *Numerical transient analysis of Markov models* // *Computers and Operations Research*. 1988. Vol. 15. No. 1. P. 19–36.
21. Klushin, Y. S. *Software implementation of mathematical models, methods and algorithms for estimating the time of execution of complex software complexes in multiprocessor computer systems*. *Bulletin of NU "Lviv Polytechnic"* № 905. *Computer systems and networks*. Lviv: NU "LP", 2018.
22. 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"* No. 881. *Computer systems and networks*. Lviv: NU "LP", 2017.

ASSESSMENT OF RELIABILITY OF PARALLEL COMPUTER SYSTEMS AT PERFORMANCE OF THE CHOSEN COMPLEX OF RELATED WORKS

Y. Klushyn

Lviv Polytechnic National University,
Computer Engineering Department

© Klushyn Y., 2019

When designing a parallel computing system for completing a given set of interactions, the robot is an important characteristic of social security. As well as the basic indicators of the higher number of computing systems, to protect yourself from technical problems, the main indicator of the higher value of the system is to assess the accuracy of the parallel calculation system of specific tasks. Tsyra otsinka seamlessly due to the effectiveness of vikoristannya obzummyudlyavnyh systems.

Key words: parallel computing systems, complex of interconnected works, direct stochastic modeling, Markov process, function of distribution of random variable, system reliability, failure.