УДК 519.718

Д.т.н., професор Гроль В.В., аспірант Фесенюк А.П., студент Суліменко Д.О.

Національний технічний університет України «Київський політехнічний інститут»

ОЦІНКА СКЛАДНОСТІ АЛГОРИТМУ РОЗРАХУНКУ НАДІЙНОСТІ ВБС НА ОСНОВІ ВИКОНАННЯ СТАТИСТИЧНИХ ЕКСПЕРЕМЕНТІВ З *GL*-МОДЕЛЯМИ

Abstract

V. V. Grol, Doct. of Science; A. P. Feseniuk, postgraduate; D. O. Sulimenko, student

In this paper one of the simulative algorithms of fault-tolerant computer systems reliability calculation is analyzed. Its complexity is also studied. The algorithm is based on executing of statistical experiments with GL-models. The algorithms of constant weight vectors generation and graph coherency check and their complexity are also given.

Вступ

Відмовостійкі багатопроцесорні системи (ВБС) знаходять широке розповсюдження в сфері управління складними об’єктами [1, 2].

Як і в [3], ВБС, що складається з n процесорів та зберігає роботоздатність при появі відмов її процесорів, кратність яких не перевищує велечини m, називатимемо базовою та позначатимемо $K(m,n)$. В [3] описана математична модель, що називається GL-моделлю, яка здатна відображати стан ВБС (роботоздатна чи ні) при появі відмов її компонент.

GL-модель ВБС, що складається з n елементів, являє собою неорієнтований циклічний граф G, де кожному його ребру приписана булева функція, аргументами якої є індикаторні змінні x_i ($i = 1 , \dots , n$), що можуть дорівнювати 1 чи 0, в залежності від стану відповідного модуля: роботоздатний чи ні. Ребро видаляється з графу *GL*-моделі, якщо відповідна йому реберна функція приймає значення 0. Зв’язність графа моделює роботоздатність ВБС. Двійковий вектор, компонентами якого є конкретні значення змінних x_i в подальшому називатимемо вектором стану ВБС.

Особливу практичну цінність становлять моделі так званих небазових ВБС, тобто стійких до деякої множини відмов, кратність яких відрізняється від m в той чи інший бік. Моделі для небазових ВБС можут бути побудовані шляхом перетворення базових [4]. Один із способів такого перетворення – введення додаткових внутрішніх ребер зі своїми функціями, що блокують втрату зв'язності графа *GL*-моделі при появі відповідних векторів стану системи.

Постановка задачі

Ймовірність відмови ВБС в часі визначається шляхом виконання статистичного експеременту з *GL*-моделями на звичайних ПК, тому доцільно оцінити його складність задля встановлення часу його виконання.

Сам алгоритм розрахунку надійності (ймовірності безвідмовної роботи) ВБС на основі статистичного експеременту можна розбити на наступні етапи:

1. Обчислення кількості повторів експеременту, потрібної для досягнення заданої точності.
2. Генерація псевдовипадкового вектору як вектору стану системи.
3. Розрахунок всіх реберних функцій, використовуючи псевдовипадковий вектор.
4. Перевірка зв'язності графа.
5. Розрахунок ймовірності реальної появи даного вектора.
6. Розрахунок ймовірності безвідмовної роботи системи.
7. Оцінка похибки.

Пункти 2-5 потрібно виконати L разів.

Оцінка складності алгоритму

Базуючись на результатах роботи [5], можна показати, що залежність кількості необхідних повторів експеременту L від заданої точності ε має вигляд:

$$L \le \frac{9N}{\varepsilon^2} \sum_{\forall \mathbf{X}} p^2(\mathbf{X}),$$

де N - кількість векторів стану системи.

Найбільшу складність для обчислення викликає частина цього виразу $\sum_{\forall \mathbf{X}} p^2(\mathbf{X})$.

В роботі [6] описано алгоритм, що дозволяє обчислити таку суму за $]\frac{7n^2}{2}+\frac{5n}{2}[$ операцій, з яких $2n^2+2n$ операцій множення, а інші – додавання. Тобто для обчислення L треба виконати $]\frac{7n^2}{2}+\frac{5n}{2}[+4$ операцій.

Генерація нового рівновагового випадкового вектору на основі попереднього виконується згідно алгоритму:

1. Звернутися до функції *rand* $n/max(rand)$ разів, де $max(rand)$ – довжина вектору, що генерується функцією *rand* за 1 звертання і в сучасних процесорах дорівнює 12, та за допомогую зсувів заповнити цими значеннями n-розрядне слово.
2. Проаналізувати згенероване слово. Якщо значення біту дорівнює 1, зсунути в попередньому векторі біт з таким же номером на позицію, на якій знаходиться наступний одиничний біт в згенерованому слові. Таку операцію провести для всіх одиничних бітів.

Для виконання такого алгоритму потрібно $2\frac{3}{4}n$ операцій, з яких $\frac{1}{12}n$ операцій звертання до функції *rand*, що займають приблизно стільки ж машинного часу, як і операція ділення.

Для обчислення всіх реберних функцій, включаючи функції 1-2 додаткових ребер, потрібно $n(m+\log(n))$ операцій множення.

Щоб визначити, чи є модель базового графа зв'язною, потрібно лише перевірити, чи не зникло 2 (або більше) його ребра. У випадку, якщо ми маємо небазову модель і додаткове ребро не зникло, то зв'язність можна перевірити за допомогою наступного алгоритму:

1. Розділити всі вершини графа на 3 різновиди: вершини, про які нічого не відомо; вершини, про які відомо, що їх можна досягти з початкової вершини, але вони не були оброблені; вершини, про які відомо, що їх можна досягти з початкової вершини та вони були оброблені.
2. Початкова розмітка: помітити всі вершини першим маркером та обрати першу вершину, яку помітити другим маркером.
3. Розмітка сусідніх вершин:

   1) якщо немає вершин, помічених другим маркером – перейти до 4-го етапу;

   2) обрати будь-яку вершину, що помічена другим маркером. Помітити її третім маркером. Всі вершини, що з'єднані з даною та помічені першим маркером, помітити другим маркером;

   3) повторити з підпункту 1.

4. Завершення роботи: підрахувати кількість вершин, що помічені першим маркером; якщо їх кількість дорівнює нулю, то граф зв'язний.

Для виконання такого алгоритму потрібно виконати $5n^2+4n+2$ операції. Ймовірність того, що додаткове ребро не зникне, дорівнює q^m, де q – ймовірність відмови процесора.

Тобто перевірка зв'язності буде виконуватися за $2n+3+q^m(5n^2+4n+2)$ операцій.

Розрахунок ймовірності появи згенерованого псевдовипадкового вектора може бути виконаний простим перемноженням ймовірностей відмови (якщо на відповідній позиції вектора стану стоїть 0) та безвідмовної роботи (якщо 1) всіх процесорів. Такий розрахунок буде обмежений $2n-m-1$ операціями, з яких n операцій множення, якщо не прийняти спеціальних заходів для спрощення розрахунків.

Для розрахунку ймовірності безвідмовної роботи достатньо від 1 відняти суму ймовірностей виникнення векторів, при яких система втратила роботоздатність. Таких векторів може бути максимум L.

Похибка обчислення безвідмовної роботи системи розраховується за $n(6n+7)$ операцій, з яких $n(4n+4)$ операцій множення.

Висновок

Для виконання статистичного експеременту потрібно виконати наступну кількість операцій:

$$]\frac{7n^2}{2}+\frac{5n}{2}[+4+L\cdot(n(m+\log(n))+6\frac{3}{4}n-m+3+q^m(5n^2+4n+2))+6n^2+7n$$

Оскільки $q^m\cdot n^2<1$, то кількість операцій в циклі $O(n(\log(n)))$. Складність проведення всіх експериментів матиме порядок $O(L\cdot n(\log(n)))$.

Література

1. Многоверсионные системы, технологии, проекты / В.С.Харченко, В.Я.Жихарев, В.М.Илюшко, Н.В.Нечипорук. - Харьков: Нац. Аэрокосм. Ун-т «Харьк. авиац. ин-т», 2003. – 486 с.
2. R.C. Suich, R.L. Patterson k-out-of-n:G systems: Some cost considerations // IEEE Trans. Reliability, vol. 40.- pp.259-264, Aug. 1991

3. Романкевич А.М., Карачун Л.Ф., Романкевич В.А. Графо-логические модели для анализа сложных отказоустойчивых вычислительных систем // Электронное моделирование.- 2001, т.23, №1, с.102-111.
4. Романкевич А.М., Иванов В.В., Романкевич В.А. Анализ отказоустойчивых многомодульных систем со сложным распределением отказов на основе циклических GL-моделей // Электронное моделирование.-2004, т.26, №5, с.67-81.
5. Гроль В.В., Романкевич В.А., Фесенюк А.П. Об оценке погрешности расчета надежности отказоустойчивых много процессорных систем // Радіоелектронні і комп‘ютерні системи.-№5, 2009, с.56-59
6. Kuo W., M. J. Zuo. Optimal reliability modeling: principles and applications. – John Wiley & Sons. Inc., Hoboken, New Jersey, 2003. - 559 p.