УДК 004.052.2

УВЕЛИЧЕНИЕ НАДЕЖНОСТИ ФУНКЦИОНИРОВАНИЯ МНОГОПОТОЧНЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ

Горбунов Николай Васильевич[1], Бычков Александр Викторович[2]

[1]*Кандидат технических наук, доцент;*
ГБОУ ВПО «Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: nikolai_gorbunov@mail.ru.

[2]*Студент;*
ГБОУ ВПО «Международный Университет природы, общества и человека «Дубна»,
Институт системного анализа и управления;
141980, Московская обл., г. Дубна, ул. Университетская, 19;
e-mail: avengerxiii@gmail.com.

В статье рассматривается задача повышения надежности функционирования автономных программно-аппаратных комплексов в условиях воздействия ионизирующих излучений. Предложен путь повышения надежности функционирования с использованием технологий многопоточного программирования. Получены результаты, касающиеся целесообразности развития предложенного подхода.

Ключевые слова: повышение надежности, система, многопоточность, автономность.

IMPROVE RELIABILITY OF MULTITHREADED INFORMATION PROCESSING SYSTEMS

Gorbunov Nikolai[1], Bychkov Alexander[2]

[1] *Candidate of Science in Engineering, Associate Professor;*
Dubna International University of Nature, Society and Man,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: nikolai_gorbunov@mail.ru.

[2]*Student;*
Dubna International University of Nature, Society and Man,
Institute of system analysis and management;
141980, Dubna, Moscow reg., Universitetskaya str., 19;
e-mail: avengerxiii@gmail.com.

The article is devoted to the problem of improving the reliability of autonomous software and hardware systems under the impact of ionizing radiations. Way to increase the reliability by using multithreaded programming technology. There are also results concerning the feasibility of the proposed approach.

Keywords: improve reliability, system, multithreading, autonomous.

Усложнение программных комплексов на современном этапе происходит из-за ряда предпосылок: расширения функциональных возможностей, повышения производительности и эффективности функционирования, улучшения пользовательского интерфейса, развития графического представления результатов и др. Аналогичная тенденция характерна и для программно-аппаратных комплексов. Обеспечение надежности сложных систем всегда было актуальной задачей и решалась эта задача на различных этапах разными способами.

Необходимость обеспечения повышенной надежности функционирования программно-аппаратных комплексов характерны для аппаратуры, предназначенной для длительного автономного функционирования в условиях воздействия ионизирующих излучений.

Ионизирующее излучение чаще всего вызывает разрушение программной части аппаратно-программного комплекса. Наибольшее разрушающее воздействие ионизирующее излучение оказывает на устройства памяти, в связи с чем может возникнуть непредсказуемое поведение всего комплекса, вплоть до его потери. Особо надо подчеркнуть, что под устройства памяти попадают как оперативные запоминающие устройства, так и всевозможные системы хранения информации, выполненные по Flash-технологии.

Наиболее распространенными способами повышения надежности являются: использование аппаратного дублирования, применение радиационно-стойких компонентов или их комбинация. Но эти решения не эффективны при случайных воздействиях на ход выполнения программы, когда формально комплекс остается работоспособным, но неправильно функционирующим. Для решения задачи в такой постановке требуется создание самовосстанавливающегося программного обеспечения, способного блокировать негативные воздействия без потери результатов и перезапуска всего комплекса в случае обнаружения сбоев или ошибок.

Предлагается способ решения за счет введения своеобразного программного дублирования в работу всего аппаратно-программного комплекса. Алгоритм программного дублирования подразумевает одновременное вычисление одних и тех же последовательностей исполняемого кода в различных потоках на отдельных копиях данных.

Основными предпосылками применения такого подхода послужили появление, широкое распространение и существенное удешевление многоядерных процессоров.

Для контроля процесса вычислений требуется внедрение механизма контрольных точек и транзакционного подхода к обработке отдельных этапов вычислений.

Достоинствами данного способа являются возможное удешевление и упрощение аппаратной части за счет отказа от избыточного аппаратурного дублирования или неиспользования в некоторых частях радиационно-стойких компонентов.

К недостатком такого подхода можно отнести снижение производительности всего комплекса. Но этот недостаток компенсируется быстрым прогрессом в развитии электронной элементной базы.

Для демонстрации и апробации предложенного подхода было создано приложение-симулятор, обладающее следующими характеристиками:

- вариант с двумя независимыми потоками обработки;
- симуляция возникновения вычислительных ошибок в любом из потоков;
- метод коррекции ошибок – возвращение потоками идентичных результатов принимается за верное, в противном случае идет перезапуск вычислений контрольной точки;
- контрольные точки – сохранение только последнего удачного вычисления.

Для имитации поэтапных вычислений был выбран в качестве модельной процедуры обработки данных матричный итерационный алгоритм. А именно, реализован алгоритм решения задачи Дирихле для задачи теплопроводности сеточным итерационным методом. Выбор обусловлен требованиями к объемам ОЗУ и легкостью реализации.

Постановка задачи для имитации поэтапных вычислений была сделана следующим образом: Решить численно задачу Дирихле для уравнения Лапласа:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0,$$

$$(x,y) \in D,$$

область D описывается уравнениями:

$$x = 0,\ x = 10,\ y = 0,\ y = 10,$$

$$u_{i0}^{(0)} = u_{i10}^{(0)} = u_{0j}^{(0)} = u_{10j}^{(0)} = 1$$

– начальные условия.

Точность решения была задана ε = 0.0001.

Шаг разбиения *h* по *x* и *y* идентичен и равен:

$$h = 0.1,\ h = 0.05,\ h = 0.033(3),\ h = 0.025$$

соответственно для различных экспериментов.

Для решения был выбран один из наиболее простых методов – процесс усреднения Либмана для систем. Конечно-разностные уравнения в таком случае имеют вид:

$$u_{ij}^{(k+1)} = \frac{1}{4}\left(u_{i+1,j}^{(k)} + u_{i-1,j}^{(k)} + u_{i,j+1}^{(k)} + u_{i,j-1}^{(k)}\right)(k = 0,1,2,\ldots).$$

При составлении уравнений была использована схема узлов, изображенная на рисунке 1.

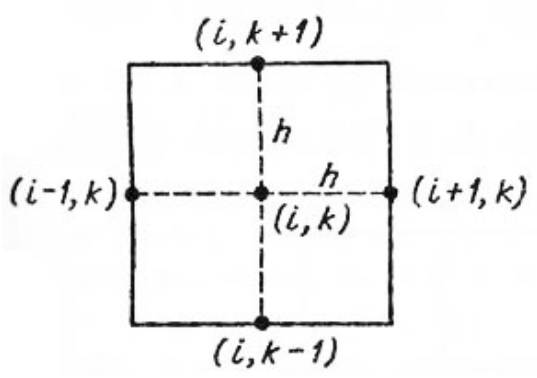

Рис. 1. Сетка, использовавшаяся при решении задачи Дирихле

Доказано, что для любого шага *h* процесс Либмана сводится к точному решению независимо от выбора начальных значений, то есть существует предел:

$$\lim_{k\to\infty} u_{ij}^{(k)} = u_{ij}.$$

Обычно итерации продолжают до тех пор, пока в двух последовательных приближениях не совпадет требуемое количество десятичных знаков. Указанный метод итерации приводит к выполнению стандартной операции усреднения в каждом внутреннем узле.

Повреждения имитировались на уровне байтовых представлений элементов матрицы. Для каждого из потоков в каждой итерации вычислений последовательность алгоритма повреждений описывается следующими этапами:

- выбирается количество областей повреждений в зависимости от числа элементов исходной матрицы;
- для каждой области выбирается один или несколько повреждаемых элементов;
- каждый из элементов преобразуется в последовательность байтов;
- в каждой из последовательностей произвольный байт заменяется на случайный;
- производится обратное преобразование из байтовых последовательностей в новые значения элементов и их запись в исходную матрицу.

Данный алгоритм обладает гибкими возможностями для управления параметрами повреждений.

Параметры алгоритма повреждений были настроены для получения вероятности повреждения в ~1% хотя бы одного значения на каждые 10000 элементов матрицы, что значительно превышает реально возможные величины, которые составляют 8 частиц на квадратный сантиметр.

Последовательность алгоритма повреждений, использовавшаяся в проведенных экспериментах, выглядела следующим образом:

1. определяется количество итераций процесса повреждений данных в матрице: на каждые полные 10000 элементов матрицы 1 итерация процесса повреждений,

2. процесс повреждений производится на каждой итерации процесса усреднения Либмана,
3. на каждой итерации процесса повреждений выбирается случайный элемент матрицы – центр области повреждений,
4. область повреждений представляет собой часть основной матрицы размером 10x10, где центральный элемент выбран на предыдущем шаге,
5. для каждого элемента области повреждений с вероятностью 0.0001 один из байтов, формирующих его, заменяется на случайный.

Таким образом, алгоритм повреждений обеспечивает имитацию повреждений на уровне физического представления, а так же локальность повреждений.

Результаты исследования работы тестового приложения на матрицах различных размерностей приведены в таблице 1.

Табл. 1. Снижение производительности для различных размеров матриц.

Размер матрицы	**Среднее число ошибочных результатов на каждые 100 вычислительных итераций**
100x100	2.1
200x200	7.9
300x300	15.6
400x400	26.8

По результатам исследований показано, что при матрицах больших размерностей при высокой вероятности повреждения данных коррекция позволяет завершить вычисления ценой производительности. Также разработанный алгоритм повреждений в процессе вычислений может быть использован в качестве компонента симулирующего повреждение кода выполняемой программы.

Список литературы

1. Рыбников Д.Л. Разработка и выбор моделей путей повышения надежности и отказоустойчивости программ реального времени бортовых многопроцессорных вычислительных систем: Автореф. дис. на соиск. учен. степ. канд. техн. наук: 05.13.13 / Моск. гос. авиац. ин-т. – М., 1997.
2. Викторова В.С. Агрегирование моделей анализа надежности и безопасности технических систем сложной структуры: Автореф. дис. на соиск. учен. степ. док. техн. наук: 05.13.01 / Ин-т пробл. упр. им. В. А. Трапезникова РАН. – М., 2009.
3. Лисс В.А. Разработка математических и имитационных моделей надежности программного обеспечения систем реального времени: Автореф. дис. на соиск. учен. степ. канд. техн. наук: 05.13.11 / СПбГЭТУ «ЛЭТИ». – СПб., 2006.