УДК 004.052.42 DOI: 10.22184/NanoRus.2019.12.89.80.81

ТЕСТИРОВАНИЕ МНОГОЯДЕРНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА ОСНОВЕ ИДЕЙ АЛГОРИТМА RSA

FUNCTIONAL AND PERFORMANCE VALIDATION OF MANY-CORE SYSTEMS BASED ON THE IDEAS OF THE RSA ALGORITHM

ЕЛИЗАРОВ СЕРГЕЙ ГЕОРГИЕВИЧ[1]
elizarov@physics.msu.ru

ЛУКЬЯНЧЕНКО ГЕОРГИЙ АЛЕКСАНДРОВИЧ[1]

МАРКОВ ДЕНИС СЕРГЕЕВИЧ[1]

МОНАХОВ АЛЕКСАНДР МИХАЙЛОВИЧ[1]

РОГАНОВ ВЛАДИМИР АЛЕКСАНДРОВИЧ[2]
radug-a@ya.ru

[1] *Физический факультет МГУ им. М. В. Ломоносова*
119991, ГСП-1, г. Москва, Ленинские горы, 1, стр. 2

[2] *Научно-исследовательский институт механики МГУ им. М. В. Ломоносова*
119192, г. Москва, Мичуринский просп., 1

ELIZAROV SERGEY G.[1]
elizarov@physics.msu.ru

LUKYANCHENKO GEORGY A.[1]

MARKOV DENIS S.[1]

MONAKHOV ALEXANDER M.[1]

ROGANOV VLADIMIR A.[2]
radug-a@ya.ru

[1] *Faculty of Physics of Lomonosov Moscow State University*
bld. 2, 1 Leninskie Gory, Moscow, 119991

[2] *Research Institute of Mechanics of Lomonosov Moscow State University*
1 Michurinsky Ave., Moscow, 119192

Предложена методика тестирования высокопроизводительных многоядерных вычислительных систем, выявляющая единичные сбои в работе аппаратуры. Методика основана на идее криптографических алгоритмов с открытым ключом, позволяет быстро проверять результаты тестирования и исключает возможность их фальсификации.
Ключевые слова: системы many-core; динамическое распараллеливание вычислений; локализация ошибок в оборудовании; алгоритм RSA.
A method for testing high-performance many-core computing systems has been proposed that identifies single faults in the operation of equipment. The method is based on the idea of cryptographic algorithms with a public key, which allows one to quickly check the results of testing and excludes the possibility of its falsification.
Keywords: many-core systems; dynamic parallelization of calculations; localization of errors in equipment; RSA algorithm.

В силу увеличения сложности вычислительных систем и создания сверхбольших специализированных вычислительных систем (системы many-core) возрастает объем ошибок и уязвимостей на аппаратном и программном уровне. Выявление ошибок и уязвимостей, тестирование являются важной задачей на различных этапах создания вычислительных систем.

Традиционным подходом к тестированию систем many-core является запуск тестов на ЭВМ «А» с последующей проверкой результатов на заведомо исправной ЭВМ «В». Однако такой подход предполагает, что имеется «образцовая» ЭВМ с производительностью, сравнимой с производительностью тестируемой вычислительной системы. В случае создания больших вычислительных систем «образцовая» ЭВМ может попросту отсутствовать. Доказательство же корректности работы на уровне математической модели устройства является сложной задачей для отработанных на настоящее время методов формальной верификации.

Одним из подходов, позволяющих решить данную проблему, является тестирование путем запуска программных тестов, обеспечивающих длительную распределенную загрузку системы с легко проверяемым результатом. По мнению авторов, тесты должны также удовлетворять следующим требованиям:

- быть простыми в реализации, осуществлять сложные преобразования при вычислении и обеспечивать быструю проверку полученных результатов;
- задействовать все основные подсистемы систем many-core, включая счетные ядра, память, коммуникационную среду;
- масштабироваться и распараллеливаться;
- иметь предсказуемое и легко варьируемое время выполнения;
- иметь высокую чувствительность к одиночным сбоям: ошибки на любом этапе выполнения программы должны с высокой вероятностью отражаться на результате теста;
- обеспечивать хорошее покрытие и псевдослучайное распределение получаемых в процессе счета промежуточных значений;
- иметь труднопредсказуемый заранее результат вычислений.

Поиск подходов к разработке тестов с позиции перечисленных выше требований целесообразно производить на основе идей криптографических алгоритмов с открытым ключом. Действительно, если взять любой шифр с открытым ключом, то дешифрование и расшифрование (последнее отличается именно наличием информации о закрытом ключе) отличаются по вычислительной сложности на много порядков. Далее будем

рассматривать наиболее известный, достаточно хорошо изученный и активно используемый на практике алгоритм RSA [1].

Рассмотрим далее методику тестирования систем many-core (рис. 1) в несколько этапов.

На первом этапе производится генерация открытого и закрытого ключа на «образцовой» ЭВМ, в качестве которой может выступать персональный компьютер. Производится выбор формулы для генерирования оснований и показателей с последующим вычислением правильного ответа для теста, полученного при помощи редукции показателей с использованием значения закрытого ключа. Общее количество вычисляемых базовых подвыражений должно быть в несколько раз больше имеющегося на тестируемой системе many-core количества счетных ядер.

На втором этапе в реализацию теста для системы many-core вносятся выбранный алгоритм генерации оснований и показателей, а также открытый ключ, после чего программа компилируется и запускается.

Заключительным этапом является сравнение полученного результата вычислений с результатом, полученным на «образцовой» ЭВМ. Если результаты совпадают, то это косвенно свидетельствует об исправности функционирования аппаратуры и системного программного обеспечения.

Разработанная методика позволяет локализовать неисправные вычислительные блоки системы many-core путем повторного тестирования, сохраняя при этом все промежуточные значения и номера блоков, на которых они вычислялись. Сравнивая промежуточные значения с аналогичными значениями, полученными на «образцовой» ЭВМ, можно обнаружить неисправные блоки.

Проверка разработанной методики осуществлялась на отечественной специализируемой системе MALT [2]. Проверка производилась на эмуляторе системы и тестовых образцах [3].

В эмулятор системы была добавлена возможность внесения одиночных ошибок в операцию умножения (для инструкций семейства MUL). В процессе выполнения теста производилось порядка 100000 операций умножения. Период внесения ошибок эмулятором соответствовал количеству операций умножения. Во всех случаях тестирование завершалось с неверным результатом, реагируя на каждую ошибку. Проведение тестирования

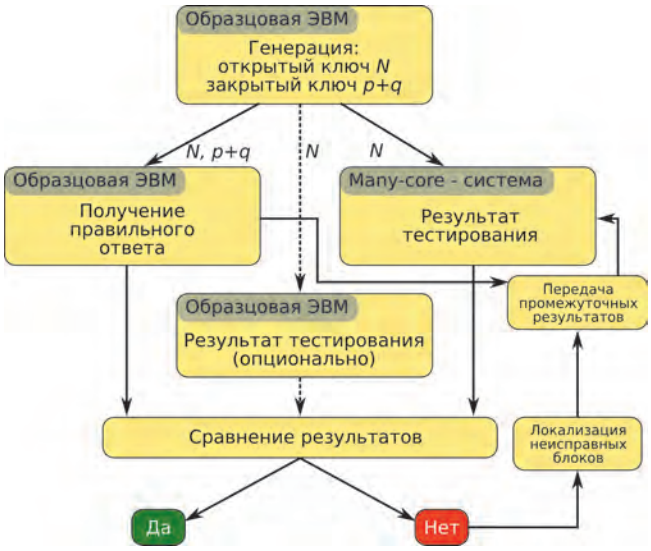

Рис. 1. Схема тестирования систем many-core

тестовых образцов системы MALT предложенной методикой показало отсутствие аппаратных и программных ошибок.

В результате проведения тестовых испытаний разработанная методика показала высокую эффективность обнаружения ошибок в системах many-core и может быть использована на различных этапах создания вычислительных систем.

ЛИТЕРАТУРА

1. Bakhtiari M., Maarof M.A. *Serious Security Weakness in RSA Cryptosystem* // International Journal of Computer Science Issues. 2012. Vol. 9. № 3. P. 175–178.
2. Елизаров С.Г., Лукьянченко Г.А., Монахов А.М., Сизов А.Д., Советов П.Н. Вычислительный модуль для многопотоковой обработки цифровых данных и способ обработки с использованием данного модуля. Патент РФ 2018.05.21. Рег. № 2018118432/08(028834).
3. С фабрики TSMC получены первые образцы микропроцессора MALT-C 9Mb96G. URL: https://maltsystem.ru/ru/news/135-fabget.