

УДК 519.86.62-192

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ ГРАФА СОСТОЯНИЙ В ЗАДАЧАХ АНАЛИЗА НАДЕЖНОСТИ ТЕХНИЧЕСКИХ СИСТЕМ

Д.Н. Шевченко, А.Ю. Литвин, М.А. Федянин

Белорусский государственный университет транспорта, Гомель

RELIABILITY ANALYSIS OF TECHNICAL SYSTEMS BY SIMULATION MODELING OF GRAPH OF STATES

D.N. Shevchenko, A.U. Litvin, M.A. Fedianin

Belarusian State University of Transport, Gomel

Рассматривается задача анализа надежности технической системы, заданной графом состояний, методом имитационного моделирования, который преодолевает ряд ограничений марковского метода анализа. Предлагается модель, программные средства и технология исследования надежности технической системы. Исследуется производительность вычислительного процесса и точность моделирования в зависимости от типа используемой вычислительной системы, ее настроек, а также от количества параллельно работающих вычислительных потоков.

Ключевые слова: надежность, граф состояний, имитационное моделирование, вычислительная система, производительность вычислительного процесса.

An approach is proposed for reliability analysis of a technical system, which is specified by the graph of states, using simulation. This method overcomes the limitations of the Markov method of analysis. The model, software and technology of investigation of the reliability of a technical system are proposed. The performance of the computational process and the accuracy of modeling, depending on the type and properties of the computer system and the number of computational flows are investigated.

Keywords: reliability, graph of states, simulation modeling, computer system, performance of computational process.

Введение

Для анализа надежности технических систем (ТС) широко используется марковский метод, который позволяет определять следующие показатели надежности восстанавливаемых, реконфигурируемых систем: вероятность безотказной работы (P) и функцию отказа (F); среднюю наработку до отказа (MTTF); среднюю наработку между отказами (MTBF); среднее время восстановления (MTTR); стационарный K и нестационарный $K(t)$ коэффициент готовности [1], [2]. В качестве математической модели марковский метод использует граф состояний (ГС) и имеет ряд ограничений:

1) время пребывания ТС в каждом состоянии должно подчиняться экспоненциальному распределению. Только в этом случае графу будет соответствовать система дифференциальных уравнений Чепмана – Колмогорова, которая имеет известное решение;

2) большое количество состояний (более 1000) не позволяет автоматизировано решать систему Чепмана – Колмогорова ни в одном из пакетов компьютерной математики.

Указанные ограничения могут преодолеваются благодаря имитационному моделированию (ИМ) ГС, т. е. компьютерному воспроизведению процесса переходов ГС в ускоренном модельном времени с одновременным сохранением статистики о переходах ГС. Основным ограничением

самого ИМ является ресурсоемкость оценки показателей высоконадежных систем: требуется большое количество реализаций модели в течение длительного модельного времени. Однако современные возможности распараллеливания вычислений в многопроцессорных компьютерах, компьютерных кластерах и облачных вычислительных сервисах снимают указанное ограничение.

1 Описание тестовой задачи

Рассмотрим задачу анализа безотказности подсистемы задающего уровня микропроцессорной централизации «ипуть» [3], включающую два резервированных АРМ диспетчера (рисунок 1.1). Для допущения об экспоненциальном распределении времени безотказной работы и времени восстановления элементов ТС марковским методом получено аналитическое решение задачи [2], а затем – решение методом ИМ.

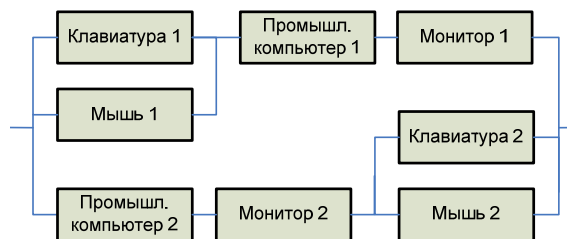


Рисунок 1.1 – Структурная схема надежности подсистемы задающего уровня микропроцессорной централизации «ипуть»

2 Предлагаемое решение задачи

Для автоматизации ИМ было разработано специальное приложение MANADA-SMGS, которое имеет следующие особенности. Приложение реализовано на Java 8 для простоты организации платформонезависимости и распараллеливания вычислений. Исходные данные о структуре и характеристиках моделируемого ГС, а также условия моделирования извлекаются из подготовленного XML-файла задания; результаты моделирования записываются в файл результатов; для отладки (при необходимости) создается лог-файл.

Исходные данные о моделируемом ГС задаются двумя матрицами:

1) матрица состояний определяет номера состояний, их принадлежность к множеству работоспособных или неработоспособных состояний ГС, вероятность пребывания в данном состоянии в нулевой момент времени;

2) матрица переходов определяет закономерности времени до перехода ГС в новое состояние. Если время до перехода в новое состояние определяется экспоненциальным распределением, то матрица переходов хранит интенсивности переходов ГС из состояния i в состояние j . Если предполагается двухпараметрическое распределение (Вейбулла, логнормальное или равномерное), то используются две матрицы. В одной из них задается первый параметр, а во второй – второй параметр распределения.

Основными показателями безотказности ГС, контролируемые в каждой реализации ИМ, являются: наработка системы до отказа (Time To Failure – TTF) или наработка между отказами (Time Between Failures – TBF), а также время восстановления ГС. В первом случае предполагается, что в начальный момент времени исследуемая ГС находится в исходном работоспособном состоянии; поэтому после моделирования отказа ГС явно переводится в исходное состояние. Выбор режима моделирования (TTF или TBF) определяется исследователем.

При запуске приложения создается вычислительный процесс, включающий поток-планировщик. В свою очередь, поток-планировщик запускает один или несколько вычислительных потоков (рисунок 2.1), которые реализуют одинаковый алгоритм, но используют генераторы случайных величин, инициализированные различными стартовыми числами. Количество и правило организации вычислительных потоков определяются пользователем. Указанная структура многопоточности не требует синхронизации потоков, что замедлило бы вычислительный процесс.

Поток-планировщик периодически проверяет условия остановки моделирования, после выполнения которых анализирует статистику, полученную каждым вычислительным потоком; формирует файл-результатов.

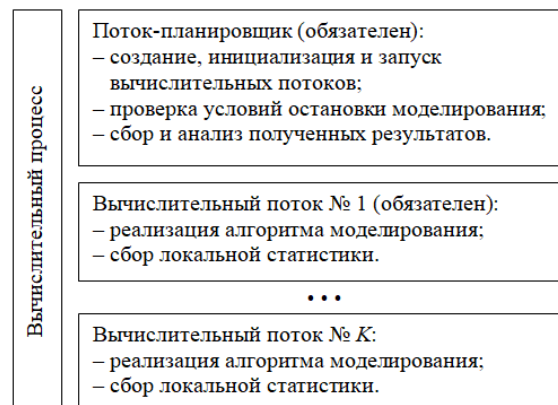


Рисунок 2.1 – Схема многопоточной организации вычислительного процесса

Каждый вычислительный поток локально сохраняет следующую статистику: сумму значений времени наработки ГС до отказа – TTF (или между отказами – TBF, в зависимости от режима моделирования); сумму квадратов значений TTF (или TBF); сумму значений времени восстановления ГС после отказа; количество реализаций TTF, превышающих заданную наработку; общее количество полученных реализаций TTF (или TBF). Кроме того, каждый поток в глобальном массиве может сохранять значения реализаций TTF, которые используются для визуализации функции отказа ГС и оценки гамма-процентной наработки до отказа.

Разработанное приложение позволяет определять:

- точечную и интервальную оценку средней наработки ГС до (или между) отказами;
- точечную оценку дисперсии TTF (или TBF);
- точечную оценку гамма-процентной наработки до отказа TTF;
- точечную и интервальную оценку вероятности безотказной работы [4].

Для обеспечения высокой точности вычислений в приложении MANADA-SMGS используются: 64-битные целые числа; 64-битные вещественные числа повышенной точности (тип DOUBLE); а также вещественные числа сверхвысокой точности для хранения 30 значащих цифр мантисы (Java-объект BigDecimal). Данной точности достаточно для того, чтобы при суммировании 10 млрд. чисел типа DOUBLE не потерять ни одной значащей цифры.

В приложении используются три качественных генератора псевдослучайных чисел: «Marsaglia-Multicarry», «XOR-Shift», «вихрь Мерсенна» [5]. Первые два алгоритма реализованы в вычислительных потоках и используются для розыгрыша времени пребывания ГС в состояниях с четными и нечетными номерами соответственно. «Вихрь Мерсенна» реализован в потоке-планировщике и используется для начальной

инициализации генераторов «Marsaglia-Multi-sargu» и «XOR-Shift» в вычислительных потоках. В свою очередь генератор «вихрь Мерсенна» может быть инициализирован значениями пользователя или таймером компьютера.

3 Исследование производительности вычислительного процесса

Производительность приложения (вычислительного процесса) определяется особенностями используемой вычислительной системы (ВС): количество ядер, тип и объем памяти, тип и настройки операционной системы и прочее, а также самого исполняемого кода (особенности работы с подсистемой ввода-вывода, взаимодействие с памятью). Выполним экспериментальное исследование влияния многопоточности на скорость ИМ при решении вышеприведенной тестовой задачи на четырех доступных ВС (с различными настройками):

- ВС1а – ноутбук Acer Extensa 5635G, 2009 г.в., Core™2 Duo CPU T6600 @2.2GHz, 2 ядра, 3 ГБ ОЗУ, Windows XP Pro SP3, 32-bit;

- ВС1б – то же, но вычислительному процессу разрешено использовать только одно из двух ядер процессора;

- ВС1в – то же, что в предыдущем пункте, но вычислительный процесс имеет наивысший приоритет «реального времени»;

- ВС2а – персональный компьютер, Pentium® CPU G840, 2013 г.в., @2.8GHz, 2 ядра, 4 ГБ ОЗУ, Windows 7 SP1, 64-bit;

- ВС2б – то же, но вычислительному процессу разрешено использовать только одно из двух ядер процессора;

- ВС2в – то же, что в предыдущем пункте, но вычислительный процесс имеет наивысший приоритет «реального времени»;

- ВС3а – сервер с 2 процессорами Xeon® CPU E5-2620 v4 @2.1GHz, 2017 г.в., Windows Server 2016 Standart 64-bit, виртуальный сервер – 4 ядра из 16 доступных;

- ВС3б – то же, но организованный виртуальный сервер использует 8 из 16 доступных ядер;

- ВС4 – облачный вычислительный сервис Amazon Elastic Compute Cloud (Amazon EC2) [6]. Amazon EC2 предоставляет в бесплатное использование виртуальный компьютер «t2.micro», который включает: 1 ГБ оперативной памяти; 10 % вычислительной мощности одного ядра процессора Intel Xeon® CPU E5-2676 v3 @2.4GHz (или аналогичного) с обеспечением постоянного уровня производительности или возможностью повышения производительности до 100 %.

Следует учитывать, что приложение MANADA-SMGS задействует лишь ресурсы процессора и не более 250 МБайт оперативной памяти; не требует дисковой памяти и сетевых устройств. Время работы приложения на ВС1, ВС2 и ВС3 задавалось равным 10 минутам, а на

ВС4 – 100 минутам (учитывая бесплатно доступную производительность процессора в 10 %). По результатам моделирования определялось количество реализаций времени наработки ТС до отказа (ТТФ, таблица 3.1).

Из значений в таблице 3.1 видно, что

- для многоядерных ВС увеличение количества потоков с 1 до 2 проявляется практически двукратным увеличением производительности;

- при увеличении количества потоков с 1 до 2 скорость моделирования существенно возрастает даже для одноядерных ВС. Это вызвано тем, что Windows не отдает единственному вычислительному потоку всех ресурсов процессора, и они могут использоваться вторым потоком;

- настройки вычислительного процесса, принятые в Windows по умолчанию (например, «средний» приоритет), предоставляют ему порядка 60–80 % ресурсов процессора;

- когда за ресурсы ядра процессора конкурируют три и более вычислительных потока, общая скорость вычислительного процесса снижается;

- максимальная производительность ядра ВС1 составляет порядка 330 реализаций ТТФ; для ВС2 и ВС3 эта цифра составляет порядка 2600 реализаций, а для ВС4 – порядка 2400 реализаций ТТФ. Используя данные значения можно прогнозировать производительность вычислительного кластера, построенного на базе множества соответствующих ВС;

- существенно низкая производительность 32-битной ВС1 (по сравнению с другими используемыми ВС) объясняется дополнительными затратами времени на операции с 64-битными числами, которыми оперирует приложение MANADA-SMGS;

- производительность ядер современного сервера ВС3 оказалась несколько ниже, чем производительность ядер ВС2. Это объясняется работой ВС3 на более низкой тактовой частоте. Кроме того, виртуальный сервер ВС3а и ВС3б – не единственный на данном физическом сервере;

- увеличение количества ядер ВС, используемых приложением MANADA-SMGS, сказывается на производительности нелинейно, ведь шины и оперативная память у вычислительных потоков – общие.

В таблице 3.2 показана сходимости результатов ИМ к точным аналитическим значениям [2] с увеличением длительности моделирования при организации трех вычислительных потоков на ВС2а. В скобках указаны доверительные интервалы для вероятности 0,95. Следует отметить, что традиционная (однопоточная) реализация приложения потребовала бы (для обеспечения аналогичной точности) в 2–2,5 раза большего времени моделирования.

Таблица 3.1 – Количество полученных реализаций ТТГ

ВС	Количество потоков									
	1	2	3	4	5	6	9	10	11	
1а	292	464	460	441	422	435	416	462	441	
1б	228	242	265	283	268	258	218	213	210	
1в	295	332	335	333	332	322	311	311	302	
2а	1580	5110	5148	5138	4976	4883	5023	4880	4524	
2б	2517	2584	2372	2474	2508	2222	2209	2265	2339	
2в	2500	2584	2636	2597	2512	2155	2552	2472	2437	
3а	1409	2952	5208	9684	10220	8819	6013	8805	9567	
3б	2031	4201	5396	6609	8882	10375	17816	18201	15342	
4	1194	2333	2392	2343	2258	2106	2154	2165	2130	

Таблица 3.2 – Сходимость оценок, полученных ИМ

Время моделирования	30 сек	10 мин	6 час	15 часов
Количество реализаций	242	4761	168302	416904
Средняя наработка до отказа, час. Точное значение – 6,94683593Е7.	6,42237475Е7 (5,57607110Е7; 7,26867840Е7)	7,05136801Е7 (6,85274954Е7; 7,24998647Е7)	6,94838475Е7 (6,91516764Е7; 6,98160185Е7)	6,93454968Е7 (6,91350517Е7; 6,95559418Е7)
Вероятность безотказной работы в течение 1 млн. часов. Точное значение – 0,9857080997.	0,9752066116 (0,946800105; 0,990798841)	0,98466708675 (0,980759147; 0,987962519)	0,98594193771 (0,985368058; 0,986498986)	0,98585765548 (0,9854946351; 0,9862138999)

Заключение

Благодаря многопоточности приложение MANADA-SMGS эффективно использует мощности доступных ВС, позволяя в несколько раз увеличить производительность по сравнению с однопоточным Windows-приложением, и преодолевает ограничения аналитического марковского метода, связанные с большой размерностью ГС и экспоненциальным временем пребывания в каждом из состояний.

Современные многоядерные ВС позволяют по-новому взглянуть на организацию имитационных экспериментов при оценке с необходимой точностью показателей высоконадежных ТС. Особенно перспективным для проведения ресурсоемких имитационных экспериментов видится использование облачных вычислительных сервисов, которые имеют следующие достоинства: ВС используется лишь по необходимости, удаленно; аппаратное и системное программное обеспечение ВС не нужно приобретать и обслуживать; в вычислительном процессе могут быть задействованы сотни процессоров; мощности сервиса постоянно возрастают; в ряде случаев допустим бесплатный уровень использования.

ЛИТЕРАТУРА

1. ГОСТ Р 51901.15–2005 (МЭК 61165:1995) Издания. Менеджмент риска. Применение марковских методов. – Введ. 2006–02–01. – М.: Стандартинформ, 2005. – 15 с.

2. Шевченко, Д.Н. Основы теории надежности: учеб.-метод. пособие для студ. техн. спец. / Д.Н. Шевченко; под ред. Л.А. Сосновского; М-во образования Респ. Беларусь, Белорус. гос. ун-т трансп. – Гомель: БелГУТ, 2010. – 250 с.

3. Бочков, К.А. Сравнительный анализ надежности эксплуатируемых на Белорусской железной дороге микропроцессорных централизаций / К.А. Бочков, С.Н. Харлап, Д.Н. Шевченко // Вестник Белорусского государственного ун-та транспорта: Наука и транспорт. – 2009. – № 2 (19). – С. 41–46.

4. Кобзарь, А.И. Прикладная математическая статистика: для инженеров и научных работников / А.И. Кобзарь. – М.: Физматлит, 2006. – 813 с.

5. Шевченко, Д.Н. Методика тестирования и использования генераторов псевдослучайных последовательностей / Д.Н. Шевченко, С.В. Кривенков // Проблемы физики, математики и техники. – 2014. – № 2 (19). – С. 89–95.

6. Документация Amazon Elastic Compute Cloud [Электронный ресурс] / Amazon Web Services, Inc. или ее дочерние организации, 2017 г. – Режим доступа: <https://aws.amazon.com/ru/documentation/ec2/> – Дата доступа: 15.02.2018.

Поступила в редакцию 23.04.18.