

Все процедуры чтения к областям данных являются длительными, поскольку сопряжены с поиском необходимой информации. А все процедуры записи – короткими, поскольку модифицируют уже найденный элемент или дописывают в конец области новый. Блокировка структуры потоком предусматривает следующие действия:

- если структура заблокирована другим потоком, то переход в состояние ожидания пока структура не будет разблокирована, т.е. пока другой поток не закончит с ней работу;
- блокировка структуры для любого другого потока;
- работа со структурой – запись или чтение;
- разблокировка структуры.

Блокировка структуры необходима в том случае, если возможно чтение или запись данных из структуры одним потоком, в то время как другой поток может осуществлять запись в данную структуру. Блокировки порождают простои в работе потоков, поэтому должны сводиться к минимуму.

Библиографические ссылки

1. Crane D., McCarthy P. Comet and Reverse Ajax: The Next-Generation Ajax 2.0, 2008. – 142 с.
2. Hypertext Transfer Protocol -- HTTP/1.1. Июнь 1999. – URL: <http://www.w3.org/Protocols/rfc2616/rfc2616.html> (дата обращения: 18.02.2011).
3. Снейдер Й. Эффективное программирование TCP/IP. Библиотека программиста. – СПб. : Питер, 2002. – 320 с. : ил.

M. V. Tyulkin, Post-graduate, Perm National Research Polytechnic University

I. V. Kapper, Perm Printing Factory (branch) of the Federal State Unitary Enterprise "GOZNAK"

E. L. Krotova, PhD (Physics and Mathematics), Associate Professor, Perm National Research Polytechnic University

L. N. Krotov, DSc (Physics and Mathematics), Professor, Perm National Research Polytechnic University

Architecture Development and Organization of Data Flows in Comet-Servers for Comet Web-Applications with "Web-Socket Streaming" Interaction Scheme. Part 1 – Description of Comet-Server

The paper offers a highlight of the main constituent elements of Comet-Server, such as various data structures and processing computation flows. Different architectural solutions are also proposed on interaction between these elements through organization of data flows depending on tasks solved by Comet-server under high loads.

Key words: software architecture, high-loaded application, data exchange, computation flow, Comet-server, Socket-server, Socket, Berkeley sockets, client-server application.

УДК 681.32

С. Ф. Тюрин, доктор технических наук, профессор, Пермский национальный исследовательский политехнический университет

О. А. Громов, аспирант, Пермский национальный исследовательский политехнический университет

А. Н. Каменских, студент, Пермский национальный исследовательский политехнический университет

ПРОГРАММНЫЙ КОМПЛЕКС ИССЛЕДОВАНИЯ МЕТОДОВ ПОВЫШЕНИЯ НАДЕЖНОСТИ

Предлагается алгоритм поиска оптимальной структурной схемы надежности системы. Приводится пример расчета структурной схемы надежности. Описывается программа, которая позволяет реализовать данный алгоритм и исследовать возможные способы повышения надежности системы для заданных параметров.

Ключевые слова: структурная схема надежности, метод наискорейшего спуска, мажоритирование, резервирование, функционально полный толерантный элемент.

При синтезе современных сложных информационно-управляющих систем всегда возникает вопрос выбора оптимальной структуры. При этом проектируемая система должна реализовывать все функции, заданные в техническом задании, и иметь минимальную стоимость, чтобы обеспечить конкурентоспособность устройства. Задача выбора оптимальной структуры также остро стоит для проектировщиков систем специального назначения. В по-

добных системах большое значение имеет надежность системы, и поэтому в процессе разработки новых устройств как правило приходится увеличивать количество аппаратуры, чтобы обеспечить нужную наработку на отказ. Однако увеличение объема аппаратуры приводит к ухудшению многих других параметров, например, таких, как стоимость устройства, мощность, рассеиваемая устройством, массогабаритные характеристики и т. д.

Однако в настоящее время процесс выбора оптимальной структурной схемы надежности не автоматизирован, и разработчикам приходится как правило вручную решать поставленную задачу. При этом велика вероятность ошибок и сложность производимых расчетов, так как подобные вычисления проводятся итерационным способом, что является довольно трудоемким. Поэтому мы предлагаем программу, которая поможет автоматизировать процесс расчета, что сможет сильно упростить решаемую задачу.

Применяемые в настоящее время методы синтеза оптимальной структурной схемы надежности представлены в справочных изданиях, например в [1, 2]. Анализ литературы показывает, что существуют упрощенные методики синтеза структурной схемы надежности системы. Так, в [3, 4] представлен метод на основе процедуры наискорейшего спуска. Недостатком указанного метода является решение поставленной проблемы только для задачи минимизации стоимости системы при заданной вероятности безотказной работы.

Мы предлагаем адаптировать данную методику и решать так же задачу максимизации вероятности безотказной работы при заданных ограничениях на стоимость системы.

В качестве математического аппарата решения задачи оптимизации будем использовать метод градиентного спуска, суть которого заключается в поиске минимума (максимума) целевой функции с помощью движения вдоль градиента, который, как известно, является направлением наискорейшего роста функции. При этом на каждом шаге итерации поиск точки максимума происходит вдоль «наилучшего» в некотором смысле направления.

Упрощенная методика синтеза оптимальной структурной схемы надежности

Пусть система состоит из n подсистем. При этом заданы значения вероятности безотказной работы P_i и стоимости W_i (где $i = 1, \dots, n$) каждой подсистемы. Имеется m методов повышения вероятности безотказной работы (ВБР), среди них могут быть мажоритарное резервирование, резервирование замещением при нагруженном резерве («горячее»), при облегченном резерве («теплое»), при холодном резерве («холодное»), скользящее резервирование с восстановлением и др. Кроме того вводятся ограничения на применение этих методов. Тогда оптимальная структура схема надежности может быть представлена в виде вектора

$$(k_i \cdot \eta), \quad (1)$$

где $i = \overline{1, n}$, а $\eta = \overline{1, m}$.

Постановка задачи для минимизации стоимости функции выглядит следующим образом: найти вектор $(k_i \cdot \eta)$, удовлетворяющий следующим условиям: $W_C \rightarrow \min$ при $P_C(t) \geq P_C^{\text{зад}}(t)$.

Постановка задачи для максимизации вероятности безотказной работы имеет вид: найти вектор

$(k_i \cdot \eta)$, удовлетворяющий следующим условиям: $P_C(t) \rightarrow \max$ при $W_C \leq W_C^{\text{зад}}$.

При этом параметр $P_C(t)$ определяется по формуле

$$P_C(t) = \prod_{i=1}^n P_i(t). \quad (2)$$

Методика включает в себя два этапа. На первом нужно убедиться, что $P_i(t) \geq P_C^{\text{зад}}(t)$. Если

$\prod_{i=1}^n P_i(t) \leq P_C^{\text{зад}}(t)$, то переходят ко второму этапу,

который заключается в увеличении резервов. Процедура итеративного увеличения резерва происходит с помощью нахождения наибольшего приращения функции в соответствии с выражением для градиентной функции. Так, для решения первой задачи поиск решения идет в соответствии с выражением

$$(\delta_i^j)^* = \max \{ \delta_i^j \}, \quad (3)$$

где $i = \overline{1, 5}$. В свою очередь, δ_i^j определяется по формуле

$$\delta_i^j = \frac{P_i^{j+1}(t) - P_i^j(t)}{W_i \cdot P_i^{j+1}(t)}, \quad (4)$$

где j – номер итерации, начинающийся с нуля.

Программа автоматизированного поиска оптимальной структурной схемы надежности

Программа автоматизированного расчета должна выполнять следующие действия.

- Сбор входных данных.
 - Синтез структурной схемы и вычисление параметров системы.
 - Вывод данных пользователю.
- Указанная программа должна реализовать расчет структурной схемы надежности следующими методами:
- Обычный метод резервирования.
 - Метод резервирования с учетом мажоритарного.
 - Метод с резервированием мажоритарного.
 - Скользящее резервирование с мажоритированием.
 - Скользящее резервирование с восстановлением.
 - Резервирование мажоритарного.

Программа работает в соответствии с блок-схемой алгоритма, представленной на рис. 1.

Опишем подробнее каждый этап работы алгоритма.

1. Загрузка начальных параметров. На данном этапе производится сбор сведений о системе. Задаются такие параметры, как количество подсистем, минимально допустимая вероятность безотказной работы, максимальная стоимость системы в элементах.

2. Загрузка параметров подсистем. На этом этапе задаются стоимости и вероятности безотказной работы каждой подсистемы.

3. Выбор варьируемой переменной. На этом шаге определяется, какую из двух поставленных задач необходимо решить: найти максимальное значение вероятности безотказной работы или найти минимальную стоимость системы при заданных ограничениях на вероятность безотказной работы.

4. Фиксирование начальных условий – это выбор значений системы для нулевого шага итерации.

5. При поиске максимального значения вероятности безотказной работы. Выполняется приращение вероятности безотказной работы для каждой подсистемы до тех пор, пока она не примет максимальное значение при заданных ограничениях на стоимость системы.

6. Нарращивание ВБР системы. На этом этапе происходит расчет вероятности безотказной работы при увеличенном значении количества резервных элементов каждой подсистемы, нахождение разницы между полученным расчетным значением и значением, полученным на предыдущем шаге. Далее происходит выбор направления наискорейшего роста целевой функции. Наконец происходит изменение начальных установок, которые были на входе алгоритма в начале данного итерационного этапа. Данный процесс повторяется до тех пор, пока рассчитываемая переменная не достигнет нужного значения.

7. Нарращивание стоимости системы по градиенту до требуемого значения. Данный этап выполняется в несколько шагов:

- Расчет вероятности безотказной работы после резервирования всей системы.
- Нахождение приращения вероятности безотказной работы.
- Выбор самого эффективного участка.
- Внесение изменений в начальные условия.

Процедура повторяется до достижения заданного максимального значения стоимости

8. Вывод результатов. Это заключительный этап алгоритма, на котором пользователь может получить результат расчета программы. Пример вывода результатов расчета показан на рис. 2.

Особо хочется отметить, что программа позволяет рассчитывать структурную схему надежности с применением скользящего резервирования, который по сути является адаптивным методом расчета. При применении скользящего резервирования с восстановлением расчет ведется по формуле

$$P_{CCCP}(t) = \left(\sum_{i=n}^{n+m} C_{n+m}^i \cdot e^{-i \cdot \lambda \cdot t} \cdot [1 - e^{-\lambda \cdot t}]^{n+m-i} + \sum_{j=1}^{\lfloor m/r \rfloor} C_{n+m}^{m+j} \cdot e^{-(n-j) \cdot \lambda \cdot t} \cdot [1 - e^{-\lambda \cdot t}]^{m-j} \right) \cdot e^{-(\lambda_{ny} + \lambda_e) \cdot t} \quad (8)$$

Данная методика расчета предполагает восстановление работоспособности подсистемы, как с помощью резервных элементов, так и с помощью восстановления элементов с помощью остаточного базиса, который образуется в случае применения функционально-полных толерантных (ФПТ) элементов. Подробнее концепция ФПТ элемента представлена в [6, 7].



Рис. 1. Блок-схема алгоритма работы программы



Рис. 2. Графическое отображение результатов расчета

Выводы

Таким образом, в представленной статье описывается упрощенная методика поиска оптимальной структурной схемы надежности на основе метода наискорейшего спуска. Приведен пример, который наглядно иллюстрирует алгоритм расчета. В статье описана программа автоматизированного синтеза систем с оптимальной структурной схемой надежно-

сти, которая может применяться как инженерами для поиска наилучших конфигураций системы, так и учеными, занимающимися исследованием методов повышения надежности систем. В дальнейшем предполагается расширить функционал программы. В частности расширить спектр методов повышения надежности, реализовать возможность построения графиков, наглядно демонстрирующих выигрыш, получаемый при введении избыточных элементов.

Библиографические ссылки

1. Надежность и эффективность в технике : справочник. В 10 т. / ред. совет: В. С. Авдуевский (пред.) и др. – Т. 1. Методология. Организация. Терминология ; под ред. А. И. Рембезы. – М. : Машиностроение, 1989. – 224 с.
2. Надежность и эффективность в технике : справочник. В 10 т. / ред. совет: В. С. Авдуевский (пред.) и др. – Т. 2. Математические методы в теории надежности и эффективности ; под ред. Б. В. Гнеденко. – М. : Машиностроение, 1987. – 280 с.
3. Основи надійності цифрових систем. Підручник / за ред. В. С. Харченка, В. Я. Жихарева. – Харків : Міністерство освіти та науки, 2004. – 572 с.
4. Харченко В. С., Лысенко И. В. Надежность, контроль и диагностика ЭВМ : метод. пособие. – ХАИ, 2001. – 65 с.
5. Тюрин С. Ф. Функционально-полные толерантные булевы функции // Наука и технология в России. – 1998. – № 4. – С. 710.
6. Тюрин С. Ф. Проблема сохранения функциональной полноты булевых функций при «отказах» аргументов // Автоматика и телемеханика. – 1999. – № 9. – С. 76–186.

S. F. Tyurin, DSc in Engineering, Professor, Perm National Research Polytechnic University

O. A. Gromov, Post-graduate, Perm National Research Polytechnic University

A. N. Kamenskikh, Student, Perm National Research Polytechnic University

Software package for investigation of reliability increase methods

The paper proposes the algorithm of searching an optimal structural scheme of a reliability system. Example of reliability structural scheme analysis is given. Software is described that allows implementing this algorithm and investigating possible methods of increasing the system reliability for assigned parameters.

Key words: reliability structural scheme, steepest descent method, majorization, redundancy, full-function tolerant element.