

Модель и алгоритм планирования параллельной обработки информации в отказоустойчивой бортовой вычислительной системе на основе временной избыточности вычислительного процесса

С. С. Зыкова

Военно-космическая академия имени А. Ф. Можайского
Санкт-Петербург, Россия
swetlanca.zykova@yandex.ru

Аннотация. Представлена модель параллельного вычислительного процесса с временной избыточностью в отказоустойчивой бортовой вычислительной системе при внешних воздействиях. Описан вычислительный процесс, включающий периодическое сохранение текущего состояния вычислений с возможностью их восстановления.

Решена частная оптимизационная задача выбора периодичности сохранения текущего состояния вычислительного процесса в условиях возможных внешних воздействий.

Описан алгоритм планирования параллельной обработки информации в отказоустойчивой БВС с периодическим сохранением состояния вычислительного процесса и его восстановлением при деградации БВС.

Представлена имитационная модель функционирования отказоустойчивой БВС при решении целевых задач в условиях внешних воздействий, приведен анализ результатов моделирования параллельной обработки данных.

Ключевые слова: отказоустойчивая бортовая вычислительная система, параллельная обработка информации, временная избыточность.

ВВЕДЕНИЕ

Большую роль в процессах сбора и обработки целевой информации в автономных комплексах играют бортовые вычислительные системы (БВС). Анализ современных БВС показал, что сейчас их возможности по высокопроизводительной обработке информации весьма ограничены. Однако новые проекты бортовых комплексов, построенных на сетевых технологиях, позволяют наращивать вычислительную мощность бортовых компьютеров на основе реализации на них параллельных вычислений.

Перспективы развития средств вычислительной техники, интеллектуализации бортовых систем свидетельствуют о целесообразности и возможности в перспективе реализации бортовой обработки данных на основе технологий параллельных вычислений [1]. Причем при решении на борту задач могут быть одинаково эффективно реализованы как параллелизм данных, так и параллелизм задач. При этом возникает новый класс научных задач планирования параллельных вычислений в условиях внешних воздействий на БВС.

Вопросы методологии организации отказоустойчивых параллельных вычислений в вычислительных системах в

условиях их деградации являются новыми и лежат на стыке теории расписаний, теории вычислительных систем, теории параллельных вычислений, теории надежности и теории вероятностей.

Концепция работы предполагает перевод процессов функционирования БВС в угрожаемый период, предполагающий возможность частичного разрушения БВС в результате внешних воздействий, в особый режим, который заключается в превентивном планировании и реализации параллельных вычислений с периодическим сохранением их состояния на общем внешнем носителе на основе временной избыточности вычислительного процесса.

Основным показателем качества функционирования БВС в этих условиях является остаточная производительность функционирования БВС при внешних воздействиях — количество выполненных БВС целевых задач за заданное время при воздействиях, не предусмотренных условиями эксплуатации, приведших к деградации БВС (отказом части вычислительных модулей).

Штатное решение целевых задач может быть прервано (нештатно завершено) внешним воздействием. Предполагается такая организация вычислений, при которой задачи, спланированные на вышедшие из строя вычислительные модули (ВМ), выполняются на исправных ВМ либо с самого их начала, либо с момента, определяемого созданием последней «контрольной точки» вычислительного процесса прерванной задачи.

На рисунке 1 показан пример параллельного выполнения четырех задач на двух ВМ БВС. На рисунке 1, *а* представлен стандартный план вычислительного процесса, на рисунке 1, *б* — тот же план, но с созданием двух «контрольных точек» для задач 1 и 4 и по одной — для задач 2 и 3. Временные затраты на создание «контрольных точек», определяющие временную избыточность решения задач, представлены заштрихованной областью после завершения каждой задачи. В случае отказа первого ВМ в момент времени, изображенный стрелкой, выполнение задач 1 и 3 будет реализовано на исправном втором ВМ после завершения на нем задач 2 и 4.

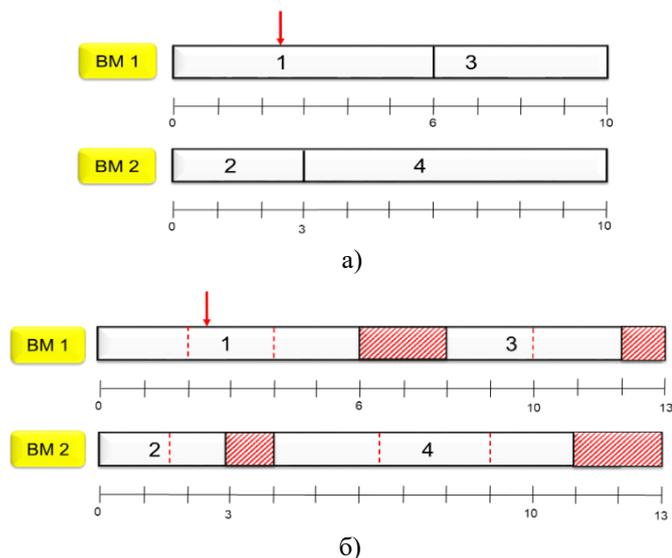


Рис. 1. Схема вычислительного процесса
 а — стандартный план вычислительного процесса;
 б — план вычислительного процесса с «контрольными точками»

При этом прерванная отказом ВМ задача 1 в стандартном варианте (рис. 2, а) будет выполняться с начала, а в предлагаемом варианте (рис. 2, б) — с момента создания ее первой «контрольной точки», что позволит сократить время на выполнение этой задачи на втором ВМ на две единицы времени.

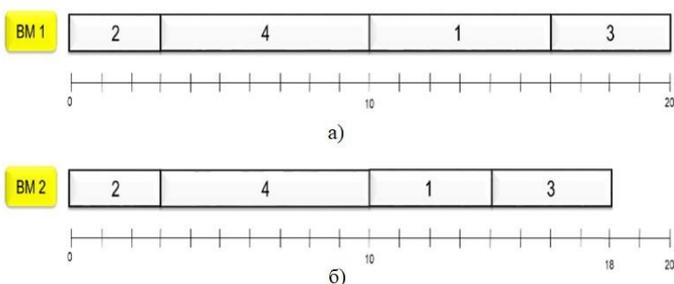


Рис. 2. Схема вычислительного процесса после отказа первого ВМ
 а — вычислительный процесс без временного резервирования;
 б — план вычислительного процесса с временным резервированием

Таким образом, за счет определенной временной избыточности, вводимой при решении каждой задачи, можно минимизировать общее время ее решения при отказе основного ВМ, на котором она выполнялась, и завершении ее выполнения на другом ВМ.

Преимущество такого подхода зависит от количества создаваемых при решении каждой задачи «контрольных точек» и временных затрат на этот процесс, а также момента времени отказа вычислительного модуля БВС.

Постановка задачи планирования параллельной обработки информации в отказоустойчивой БВС на основе временной избыточности вычислительного процесса

По заданным параметрам задач, параметрам БВС, параметрам возможного воздействия на БВС требуется найти

план (расписание) вычислительного процесса, обеспечивающий максимальное значение остаточной производительности функционирования БВС при внешних воздействиях.

При этом будем придерживаться трех допущений:

1. Любой ВМ может выполнить любую задачу.
2. Решение целевой задачи, прерванной на одном ВМ, может быть завершено на другом ВМ.
3. Все задачи независимые, т. е. результат выполнения одной задачи не влияет на решение другой.
4. Внешнее воздействие на ВМ БВС прекращает его функционирование на период реализации плана вычислений.

Дано:

1. Множество Z задач с параметрами $\langle \tau_i \rangle, i = 1, \dots, n$, где τ_i — время решения задачи, n — количество задач.
2. Количество m вычислительных модулей.
3. Функция $F(\xi)$ распределения моментов времени ξ реализации воздействия на БВС.

Найти:

1. Зависимость $S_{\Xi}(F(\xi))$ остаточной производительности функционирования БВС от плана $\Xi(Z, m)$ вычислительного процесса и функции $F(\xi)$ распределения моментов времени ξ реализации внешнего воздействия.
2. План $\Xi = \{t_1, \dots, t_n, k_1, \dots, k_m\}$ вычислительного процесса, содержащий «контрольные точки» ($t_i, i = 1, \dots, n$ — планируемые моменты времени завершения выполнения задач; $k_i, i = 1, \dots, m$ — количество «контрольных точек»), такой, что

$$\Xi^* = \arg \max_{\Xi \in \Xi^{\text{доп}}} s_{\Xi}(F(\xi)),$$

где s_{Ξ} — остаточная производительность БВС при реализации плана Ξ ;

$\Xi^{\text{доп}}$ — множество допустимых планов вычислительного процесса.

МОДЕЛЬ ПАРАЛЛЕЛЬНОГО ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА С ВРЕМЕННОЙ ИЗБЫТОЧНОСТЬЮ В ОТКАЗОУСТОЙЧИВОЙ БВС ПРИ ВНЕШНИХ ВОЗДЕЙСТВИЯХ

Рассмотрим общую схему БВС (рис. 3), в которой несколько вычислительных модулей по заявкам на решение задач в соответствии с некоторым расписанием вычислительного процесса решают целевые задачи. С некоторой периодичностью каждый ВМ записывает на общее внешнее запоминающее устройство состояние процесса выполнения текущей программы решения задачи (содержимое рабочей памяти, общих регистров, слово состояния программы).

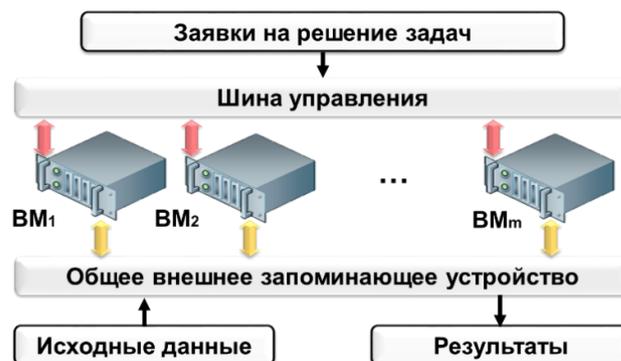


Рис. 3. Общая схема многомодульной БВС

В случае отказа некоторого ВМ спланированные для решения на нем задачи ставятся на решение после освобождения исправных ВМ. При этом те задачи, которые еще не начинали выполнение на отказавшем ВМ, будут выполнены на одном из исправных ВМ с самого их начала, а задача, выполнение которой было прервано отказом ВМ, будет довыполнена с момента создания ее последней «контрольной точки» [2].

План вычислительного процесса представлен множеством моментов времени завершения задач. При рассматриваемом подходе период решения задачи $[0, \gamma]$ состоит из k интервалов, каждый из которых, кроме последнего, включает интервал τ выполнения задачи и интервал Δ записи промежуточных результатов и состояния программы в память, то есть создания «контрольной точки» решения задачи. Предположим, что отказ одного ВМ происходит в момент времени ξ (рис. 4).

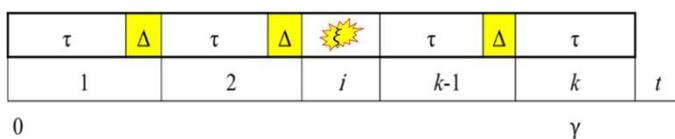


Рис. 4. Пример вычислительного процесса на одном ВМ

Таким образом, общее время решения задачи составит

$$\gamma = k\tau + (k - 1)\Delta = T + (k - 1)\Delta,$$

где T — «чистое» время, необходимое для решения задачи.

Если в момент времени ξ произошел отказ ВМ на i -м интервале решения задачи, $1 \leq i \leq k$, то в этот момент времени в памяти находится контрольная точка решения задачи, реализованного на $i - 1$ интервалах. Тогда становится возможным завершить решение задачи на другом вычислительном модуле, начиная с контрольной точки, созданной на $i - 1$ интервале.

Длительность интервала τ выполнения задачи, а следовательно, и значение времени θ выполнения прерванной задачи, зависят от величины k .

Если известна плотность распределения $f(\xi)$ момента времени отказа ВМ, то математическое ожидание $\bar{\theta}$ времени θ выполнения задачи составит

$$\bar{\theta} = \int_0^\gamma f(\xi)\theta(k, \xi)d\xi.$$

Задача состоит в оптимизации частоты создания «контрольных точек» при решении задачи с учетом вероятности отказа вычислительного модуля и возможностью ее завершения на другом ВМ.

На рисунке 5 представлено параллельное выполнение двух одинаковых по времени выполнения задач на двух ВМ, вероятность отказа каждого из которых равна q .

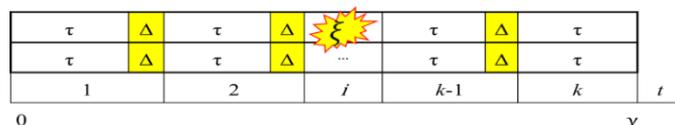


Рис. 5. Пример вычислительного процесса на двух ВМ

Обе задачи будут выполнены в случае, если откажет только один из двух ВМ или оба останутся работоспособными. Тогда среднее время, затраченное на выполнение задач, представлено выражением

$$\bar{\theta} = (1 - q)^2(T + (k - 1)\Delta) + q(1 - q)\left(3T + (k - 1)\Delta - \frac{T(T + k\Delta)(k - 1)}{k(T + (k - 1)\Delta)}\right). \quad (1)$$

Оптимальное количество «контрольных точек» для обеспечения восстановления вычислительного процесса после отказа ВМ находится из следующего уравнения, которое может быть решено как аналитически, так и численно.

$$\Delta k^2(k^2\Delta^2 + 2k\Delta T - 2k\Delta^2 - 2\Delta T + \Delta^2 + T^2) - qT^2(2k\Delta - \Delta + T) = 0.$$

При этом в общем случае требуется округление значения искомого корня уравнения до натурального числа.

АЛГОРИТМ ПЛАНИРОВАНИЯ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ ИНФОРМАЦИИ В ОТКАЗОУСТОЙЧИВОЙ БВС С ПЕРИОДИЧЕСКИМ СОХРАНЕНИЕМ СОСТОЯНИЯ ВЫЧИСЛИТЕЛЬНОГО ПРОЦЕССА И ЕГО ВОССТАНОВЛЕНИЕМ ПРИ ДЕГРАДАЦИИ БВС

Вопросам планирования параллельных вычислений в научной литературе уделено достаточно много внимания. Известно множество алгоритмов как оптимального [3, 4], так и приближенного планирования [5–7]. Применительно к рассматриваемым условиям функционирования БВС целесообразно применение приближенных алгоритмов планирования.

Идея алгоритма заключается в восстановлении прерванного из-за отказа ВМ процесса решения задачи на исправном ВМ не с самого его начала, а с последнего сохраненного состояния выполнения программы («контрольной точки») (рис. 6).

При этом целесообразно оптимизировать количество «контрольных точек», так как, с одной стороны, увеличение их количества позволяет затрачивать меньше времени на завершение вычислений прерванной отказом ВМ задачи, с другой — создание каждой «контрольной точки» вносит временную избыточность в вычислительный процесс.

Суть алгоритма: при отказе ВМ прерванная решаемая на нем задача назначается для решения с момента времени последней «контрольной точки» на исправный ВМ. Остальные, еще не решенные, задачи, назначенные на отказавший ВМ, перераспределяются на исправные модули алгоритмом FFD («первый подходящий с упорядочением») [8].

Предложенный алгоритм имеет полиномиальную сложность и может быть реализован в режиме реального времени.

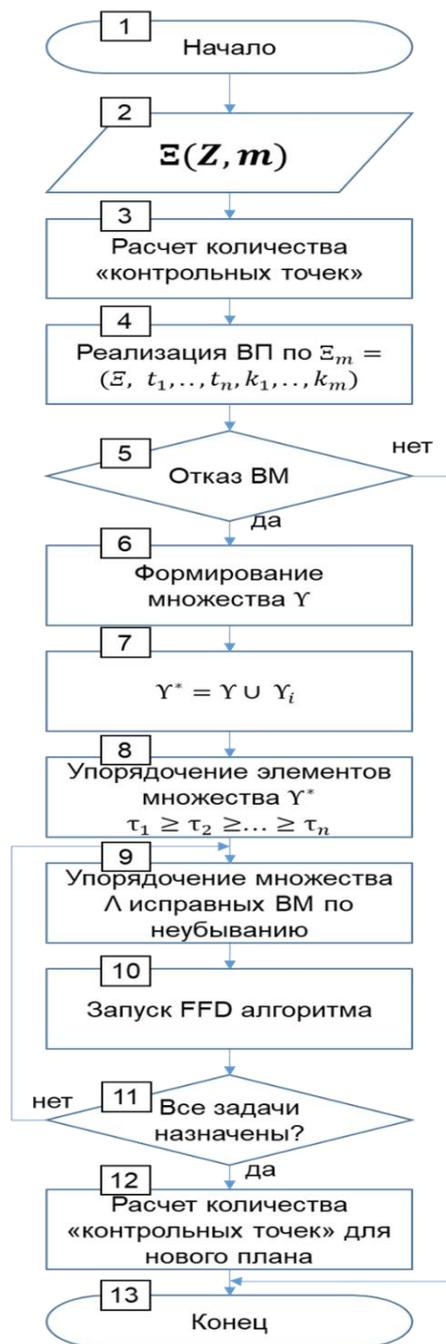


Рис. 6. Алгоритм планирования параллельной обработки информации в отказоустойчивой БВС с периодическим сохранением состояния вычислительного процесса и его восстановлением при деградации БВС

ИМИТАЦИОННАЯ МОДЕЛЬ ФУНКЦИОНИРОВАНИЯ ОТКАЗОУСТОЙЧИВОЙ БВС ПРИ РЕШЕНИИ ЦЕЛЕВЫХ ЗАДАЧ В УСЛОВИЯХ ВНЕШНИХ ВОЗДЕЙСТВИЙ

Для оценивания эффективности применения отказоустойчивой БВС с временной избыточностью реализации параллельных вычислений была построена имитационная модель процесса функционирования этой БВС.

Имитационное моделирование заключается в следующем:

1. Генерируется массив значений времени выполнения каждой задачи.

2. Строится план параллельного вычислительного процесса для заданного количества ВМ БВС.

3. Рассчитывается требуемое количество «контрольных точек» для вычислительного процесса на каждом ВМ.

4. Планируемое время вычислений на каждом ВМ увеличивается на суммарную величину значений интервалов создания «контрольных точек».

5. Случайным образом формируется номер отказавшего ВМ и момент времени его отказа.

6. Задача, выполняющаяся на отказавшем ВМ, считается прерванной в момент отказа ВМ, и ее дообработка проводится на одном из исправных ВМ с момента создания последней «контрольной точки».

7. Задачи, спланированные к выполнению на отказавшем ВМ после момента времени его отказа, переназначаются для обработки на исправные ВМ.

8. Оценивается общее время w вычислительного процесса как максимум из моментов времени завершения вычислений на каждом ВМ БВС.

9. Пункты 1–8 выполняются заданное количество раз.

10. По формуле (1) оценивается среднее время $\bar{\theta}$ вычислительного процесса.

На основе разработанных модели и алгоритма проведено имитационное моделирование, получены и исследованы зависимости, показанные на рисунках 7–9.

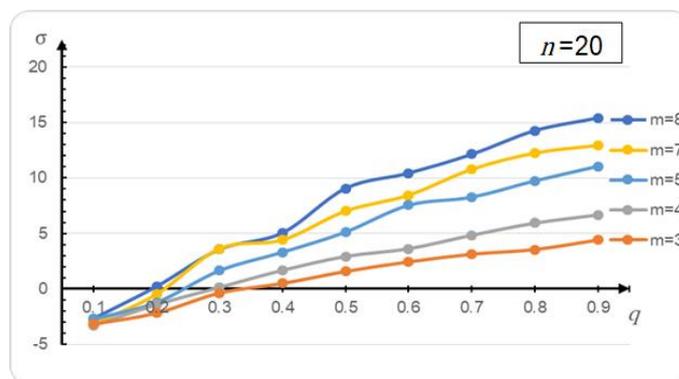


Рис. 7. Зависимость относительного выигрыша от реализации отказоустойчивого вычислительного процесса с временной избыточностью от вероятности внешнего воздействия на БВС

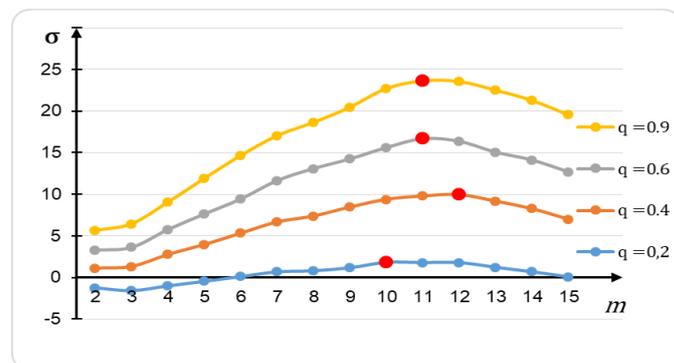


Рис. 8. Зависимость относительного выигрыша от реализации отказоустойчивого вычислительного процесса с временной избыточностью от количества ВМ в БВС

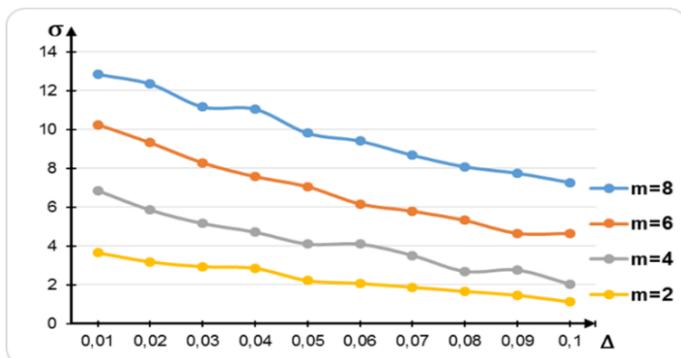


Рис. 9. Зависимость относительного выигрыша от реализации отказоустойчивого вычислительного процесса с временной избыточностью от затрат времени на создание «контрольной точки»

Представленные зависимости демонстрируют преимущество предлагаемого способа введения временной избыточности (при возрастании угрозы деградации БВС), которое увеличивается:

- с увеличением (до некоторого значения) количества ВМ;
- с увеличением вероятности ДВ на БВС;
- с сокращением временных затрат на создание «контрольных точек».

ЗАКЛЮЧЕНИЕ

Разработанная модель, в отличие от известных, учитывает временную избыточность вычислительного процесса и позволяет оценить значение показателя остаточной производительности БВС в условиях возможной ее деградации.

Представленный алгоритм предполагает проактивное сохранение состояния вычислительного процесса с возможностью его восстановления при отказах вычислительных модулей, что обеспечивает снижение потерь производительности (повышение значения остаточной производительности) при деградации вычислительной системы.

Использование предлагаемого подхода к планированию параллельных вычислений в условиях внешних воздействий позволяет снизить потери остаточной производительности БВС на величину до 15-25 %.

ЛИТЕРАТУРА

1. Кустов, В. Н. Основы теории ограниченного структурного параллелизма. — Санкт-Петербург: Министерство обороны РФ, 1992. — 246 с.
2. Метод отказоустойчивой параллельной обработки информации в бортовых вычислительных системах летательных аппаратов на основе временной избыточности вычислительного процесса / А. Г. Басыров, С. С. Зыкова, И. Н. Кошель, В. В. Кузнецов // Авиакосмическое приборостроение. 2023. № 6. С. 33–39. DOI: 10.25791/aviakosmos.6.2023.1345.
3. Crainic, T. G. Parallel Branch-and-Bound Algorithms / T. G. Crainic, B. Le Cun, C. Roucairol // Parallel Combinatorial Optimization / E.-G. Talbi (ed.). — Hoboken (NJ): Wiley-Interscience, 2006. — Pp. 1–28. — (Wiley Series on Parallel and Distributed Computing). DOI: 10.1002/9780470053928.ch1.
4. Lenstra, J. K. Sequencing by enumerative methods. — Amsterdam: Mathematisch Centrum, 1977. — 208 p. — (Mathematical Centre Tracts; Vol. 69).
5. Černý, V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm // Journal of Optimization Theory and Applications. 1985. Vol. 45, Is. 1. Pp. 41–51. DOI: 10.1007/BF00940812.
6. Michalewicz, Z. Genetic Algorithms + Data Structures = Evolution Programs. Third, Revised and Extended Edition. — Heidelberg: Springer-Verlag, 1998. — 407 p.
7. Sandnes, F. E. A Hybrid Genetic Algorithm Applied to Automatic Parallel Controller Code Generation / F. E. Sandnes, G. M. Megson // Proceedings of the Eighth Euromicro Workshop on Real-Time Systems (L'Aquila, Italy, 12–14 June 1996). — Institute of Electrical and Electronics Engineers, 1996. — Pp. 70–75. DOI: 10.1109/EMWRTS.1996.557799.
8. Adan, J. M. Meeting Hard-Real-Time Constraints Using a Client-Server Model of Interaction / J. M. Adan, M. F. Magalhães, K. Ramamritham // Proceedings of the Seventh Euromicro Workshop on Real-Time Systems (Odense, Denmark, 14–16 June 1995). — Institute of Electrical and Electronics Engineers, 1995. — Pp. 286–293. DOI: 10.1109/EMWRTS.1995.514323.

A Model and Algorithm for Planning Parallel Information Processing in a Fault-Tolerant On-Board Computing System Based on the Time Redundancy of the Computing Process

S. S. Zykova

Military Aerospace Academy
Saint Petersburg, Russia
swetlanca.zykova@yandex.ru

Abstract. A model of a parallel computing process with temporary redundancy in a fault-tolerant on-board computing system under external influences is presented. A computational process is described, including periodic preservation of the current state of calculations with the possibility of their restoration.

A particular optimization problem of choosing the periodicity of maintaining the current state of the computing process under possible external influences is solved.

An algorithm is described for planning parallel processing of information in a fault-tolerant UAV with periodic preservation of the state of the computing process and its restoration during degradation of the UAV.

A simulation model of the functioning of a fault-tolerant UAV is presented when solving target tasks under conditions of external influences, an analysis of the results of modeling parallel data processing is given.

Keywords: fault-tolerant onboard computing system, parallel processing of information, temporary redundancy.

REFERENCES

1. Kustov V. N. Fundamentals of the theory of bounded structural parallelism [Osnovy teorii ogranichennogo strukturnogo parallelizma]. Saint Petersburg, Ministry of Defense of the Russian Federation, 1992, 246 p.
2. Basyrov A. G., Zykova S. S., Koshel I. N., Kuznecov V. V. The Method of Fault-Tolerant Parallel Processing of Information in On-Board Computing Systems of Aircraft Based on the Temporary Redundancy of the Computing Process [Metod otkazoustoychivoy paralelnoy obrabotki informatsii v bortovykh vychislitelnykh sistemakh letatelnykh apparatov na osnove vremennoy izbytochnosti vychislitelnogo protsesssa], *Aerospace Instrumentation [Aviakosmicheskoe priboro-stroenie]*, 2023, No. 6, Pp. 33–39.
DOI: 10.25791/aviakosmos.6.2023.1345.

3. Crainic T. G., Le Cun B., Roucairol C. Parallel Branch-and-Bound Algorithms. In: *Talbi G. (ed.) Parallel Combinatorial Optimization*. Hoboken (NJ), Wiley-Interscience, 2006, Pp. 1–28. DOI: 10.1002/9780470053928.ch1.

4. Lenstra J. K. Sequencing by enumerative methods. Amsterdam, Mathematisch Centrum, 1977, 208 p.

5. Černý V. Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm, *Journal of Optimization Theory and Applications*, 1985, Vol. 45, Is. 1, Pp. 41–51. DOI: 10.1007/BF00940812.

6. Michalewicz Z. Genetic Algorithms + Data Structures = Evolution Programs. Third, Revised and Extended Edition. Heidelberg, Springer-Verlag, 1998, 407 p.

7. Sandnes F. E., Megson G. M. A Hybrid Genetic Algorithm Applied to Automatic Parallel Controller Code Generation, *Proceedings of the Eighth Euromicro Workshop on Real-Time Systems, L'Aquila, Italy, June 12–14, 1996*. Institute of Electrical and Electronics Engineers, 1996, Pp. 70–75. DOI: 10.1109/EMWRTS.1996.557799.

8. Adan J. M., Magalhaes M. F., Ramamritham K. Meeting Hard-Real-Time Constraints Using a Client-Server Model of Interaction, *Proceedings of the Seventh Euromicro Workshop on Real-Time Systems, Odense, Denmark, June 14–16, 1995*. Institute of Electrical and Electronics Engineers, 1995, Pp. 286–293. DOI: 10.1109/EMWRTS.1995.514323.