УДК 004.021

Методика парирования сбоев и отказов в многомодульной вычислительной системе на основе создания и репликации контрольных точек

Кочуров Денис Анатольевич — адъюнкт кафедры информационно-вычислительных систем и сетей.
Научные интересы: отказоустойчивые вычислительные системы.
E-mail: deniskochurov45@yandex.ru

Военно-космическая академия имени А. Ф. Можайского, Россия, 197198, Санкт-Петербург, ул. Ждановская, д. 13

Для цитирования: Кочуров Д. А. Методика парирования сбоев и отказов в многомодульной вычислительной системе на основе создания и репликации контрольных точек // Интеллектуальные технологии на транспорте. 2025. № 2 (42). С. 103–112. DOI: 10.20295/2413-2527-2025-242-103-111

Аннотация. *Задача по повышению оперативности обработки целевой информации требует новых подходов к возможности быстрого восстановления вычислительной системы после сбоев и отказов.* ***Цель:*** *описать методику парирования сбоев и отказов в многомодульной вычислительной системе, которая реализует периодическое сохранение состояния вычислений (контрольных точек) и обмен ими между всеми вычислительными модулями.* ***Результаты:*** *сформулирована постановка задачи планирования такого вычислительного процесса, предполагающая определение оптимального количества и моментов времени создания контрольных точек. Обоснованы моменты времени создания контрольных точек в зависимости от закона распределения моментов времени отказов вычислительных модулей.* ***Практическая значимость:*** *представлены результаты имитационного моделирования вычислений в рамках предлагаемого подхода, доказывающие целесообразность применения предлагаемой методики.*

Ключевые слова: *многомодульная вычислительная система, модель вычислительного процесса, контрольная точка*

2.3.6 — *методы и системы защиты информации, информационная безопасность (технические науки);*
1.2.2 — *математическое моделирование, численные методы и комплексы программ (технические науки)*

Введение

В настоящее время широко применяются многомодульные вычислительные системы, функционирующие в составе автоматизированных систем в условиях возмущающих факторов среды. Примерами таких систем могут быть как бортовые, так и наземные вычислительные комплексы различного назначения. Функционирование вычислительных комплексов в реальном масштабе времени создает проблему необходимости их быстрого восстановления после сбоев и отказов или парирования этих отказов на основе структурной и временной избыточности.

Известным подходом к решению этой проблемы является создание «контрольных точек» [1–3], то есть периодическое сохранение состояния программы на некоторый носитель данных. В случае отказа вычислительного модуля (ВМ) производится «откат» к ближайшей доступной контрольной точке. В восстанавливаемой вычислительной системе после отказа ее ВМ он перезапускается, и вычисления продолжаются на том же ВМ. При функционировании невосстанавливаемого ВМ созданная им до отказа контрольная точка реплицируется на один или несколько других ВМ вычис-

лительной системы, и программа, выполняемая на отказавшем ВМ, возобновляется с последней контрольной точки на исправном ВМ.

В статье рассматривается многомодульная вычислительная система, в которой каждый ВМ периодически сохраняет состояние своего активного вычислительного процесса (создает контрольную точку) и передает ее другим ВМ в рамках проактивной реакции на свой отказ. Если такой отказ происходит, то возможно восстановление прерванных вычислений с контрольной точки на другом ВМ.

В отличие от известных работ [4, 5], предлагаемый в статье подход не требует выделенного запоминающего устройства для хранения контрольных точек, не ограничивается рассмотрением одного вида распределения моментов времени деструктивных воздействий на вычислительную систему и позволяет обосновать моменты времени создания контрольных точек в зависимости от вида этого распределения.

Постановка задачи планирования репликации контрольными точками в многомодульной вычислительной системе

Задача планирования репликации контрольными точками заключается в определении оптимальных моментов времени создания контрольных точек вычислительного процесса на каждом вычислительном модуле вычислительной системы для обеспечения ее отказоустойчивости при ограничениях на время завершения выполнения целевых задач.

В соответствии с [6], отказоустойчивость — способность системы, продукта или компонента работать, как предназначено, несмотря на наличие дефектов программного обеспечения или аппаратных средств.

В рассматриваемом контексте отказоустойчивость определяется количеством вычислительных модулей, отказ которых не приведет к срыву решения целевых задач за заданное время.

Дано: вычислительная система, состоящая из m ВМ, которая должна за время T выполнить множество задач (программ). Время, затрачиваемое программами на j-м ВМ, составляет θ_j, $\theta_j \le T$. Известно, что на вычислительную систему влияют возмущающие факторы, которые обуславливают возможность отказа любого ВМ в момент времени τ, распределенный по закону $F(\tau)$. Также известны затраты $c_i(t)$ времени на создание контрольной точки на j-м ВМ, требующей объема $v_j(t)$ памяти на момент времени t.

Найти: для каждого ВМ оптимальное количество контрольных точек и моменты времени их создания для обеспечения заданного уровня d отказоустойчивости вычислительной системы при ограничениях на объем $v^{\text{доп}}$ памяти каждого ВМ, доступной для хранения контрольных точек.

Модель вычислительного процесса с контрольными точками

Пусть на интервале времени $[0, T]$ решения целевых задач в моменты времени $w_1, w_2, \ldots, w_n$ создано n контрольных точек (КТ), причем временные затраты на создание i-й контрольной точки, $1 \le i \le n$, составляют $c(w_i)$ (рис. 1).

При отсутствии КТ в случае отказа ВМ, для восстановления вычислений необходимо проводить их заново с нулевого момента времени. При наличии КТ вычисления можно восстанавливать с момента времени, соответствующего последней КТ.

Определим математическое ожидание φ_n величины сокращения времени на восстановление вычислений при использовании n контрольных точек:

$$\varphi_n = \sum_{j=1}^{n} w_i \left(1 - \alpha_i F_i\left(w_i + c\left(w_i\right)\right)\right), \qquad (1)$$

где $\alpha_i = \prod_{j=1}^{i-1}\left(1 - F_j\left(w_j + c\left(w_j\right)\right)\right)$, $\alpha_1 = 1$;

$F_i(x)$ — функция распределения случайной величины x на интервале $x > w_{i-1}$.

Величина α_i есть вероятность того, что отказ ВМ не произошел на предыдущих $j = 1, 2, \ldots, i - 1$ интервалах между моментами времени создания КТ вычислительного процесса. Если вероятность возникновения отказа после момента времени создания последней КТ не зависит от вероятности этого события на предыдущих интервалах, то значение α_i следует полагать равным 1.

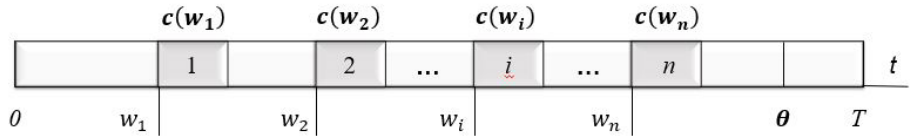

Рис. 1. Схема размещения контрольных точек

Для определения оптимальных моментов времени создания КТ найдем производные функции (1) по каждой переменной w_i, $i=\overline{1,n}$.

$$\frac{\partial \varphi_n}{\partial w_i}=1-\alpha_i\left(F_i\left(w_i+c\left(w_i\right)\right)+w_i f_i\left(w_i+c\left(w_i\right)\right)\right),$$

где $f_i(x)$ — плотность распределения случайной величины x на интервале $x > w_{i-1}$.

Тогда значения w_i можно найти, используя уравнение:

$$1-\alpha_i\left(F_i\left(w_i+c\left(w_i\right)\right)\right)+w_i f_i\left(w_i+c\left(w_i\right)\right)=0,$$

решением которого является

$$w_i=\frac{\frac{1}{\alpha_i}-F_i\left(w_i+c\left(w_i\right)\right)}{f_i\left(w_i+c\left(w_i\right)\right)}. \qquad (2)$$

Используя это выражение, например, для равномерного распределения момента времени отказа ВМ, определим, что оптимальными моментами времени создания КТ являются значения:

$$\tfrac{1}{2}T,\tfrac{3}{4}T,\tfrac{7}{8}T,\ldots,\frac{(2^i-1)}{2^i}T,\ldots .$$

Для показательного распределения с параметром λ в условиях, когда вероятность возникновения отказа после момента времени создания последней КТ не зависит от его вероятности на предыдущих интервалах α_i = const = 1, оптимальными моментами времени создания КТ являются значения $w_i = w_{i-1} + 1/\lambda$.

Решение уравнения (2) позволяет определить оптимальный момент времени для создания каждой контрольной точки для любого вида распределения момента времени отказа ВМ.

Так как функция (1) является неубывающей, то максимальное количество n^* КТ на j-м ВМ определяется максимальным значением n_j, удовлетворяющим ограничению

$$\sum_{i=1}^{n_j}c\left(w_i\right)\le T-\theta_j,\ \ \forall j\in[1,m]. \qquad (3)$$

В случае, когда $c(w_i)$ = const, значение n^* можно найти из выражения:

$$n^*=\left[\frac{T-\theta_j}{c}\right],\ \ \forall j\in[1,m],$$

где запись $[x]$ означает «ближайшее целое, меньшее или равное x».

С другой стороны, целесообразность создания очередной i-й КТ обоснована, если выполняется неравенство:

$$w_i-w_{i-1}>c\left(w_i\right). \qquad (4)$$

Среднее значение затрат времени $\overline{\theta}_j$ на выполнение вычислительного процесса на j-м ВМ с учетом возможности его восстановления с последней КТ составит:

$$\overline{\theta}_j=\theta_j+\sum_{i=1}^{n}\left(c\left(w_i\right)+\int_{w_i}^{w_{i+1}}\left(\tau-w_i\right)f_i\left(\tau\right)\partial\tau\right).$$

Таким образом, при моделировании вычислительного процесса с контрольными точками возможно оценивание уровня повышения его оперативности на основе выражения (1). Определение момента времени создания очередной КТ проводится в процессе реализации вычислительного процесса на основании выражения (2), а обоснованность создания очередной КТ формируется на основе выражений (3) и (4).

Модель вычислительного процесса с контрольными точками в многомодульной вычислительной системе

Отказоустойчивость вычислительной системы будем оценивать количеством ВМ, отказ которых не приведет к тому, что цель функционирования

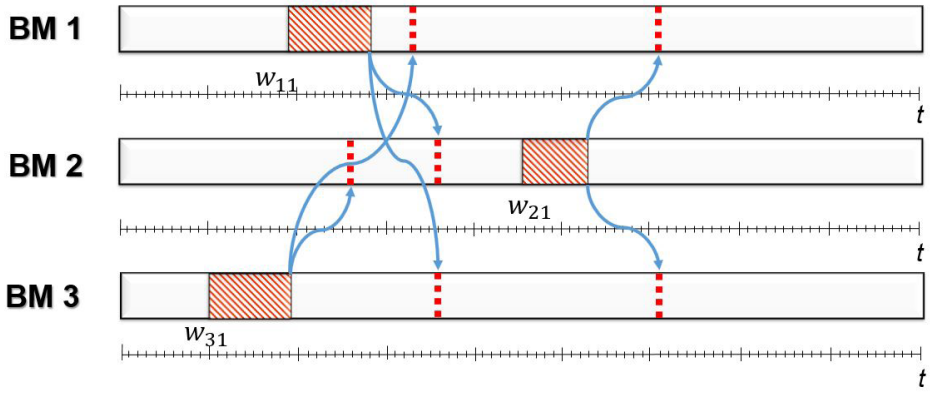

Рис. 2. Схема вычислительного процесса в вычислительной системе с репликацией контрольными точками

ВС не будет достигнута, то есть решение заданного количества задач не будет полностью завершено к директивному сроку.

Рассмотрим схему вычислительного процесса в отказоустойчивой вычислительной системе с межмашинным обменом контрольными точками (рис. 2).

Так как при отказе *j*-го ВМ выполняемый на нем вычислительный процесс может быть восстановлен с последней контрольной точки, размещенной на исправном ВМ, время завершения вычислений на последнем увеличивается на величину $\theta_j - w_{jk}$, где *k* — номер последней КТ, созданной на *j*-м ВМ и переданной на исправный ВМ.

Будем полагать, что контрольная точка, созданная на любом ВМ, передается на все остальные ВМ, что обеспечивает возможность завершить любой прерванный отказом ВМ на другом исправном ВМ с момента создания его последней КТ.

Возможность завершения всех вычислительных процессов при отказе некоторого количества μ ВМ из *m*, входящих в вычислительную систему, зависит от моментов времени τ_1, τ_2, …, τ_μ отказов ВМ, количества $m - \mu$ оставшихся исправными ВМ, резервов времени на исправных ВМ для завершения вычислительных процессов, прерванных на отказавших ВМ, моментом времени создания на каждом отказавшем ВМ последней перед отказом КТ и затратами времени на создание КТ.

После отказа очередного ВМ определяется исправный ВМ, на котором будет завершено выполнение прерванного отказом ВМ вычислительного процесса. Рациональным выбором является назначение для завершения вычислений такого ВМ, резерв времени вычислительного процесса которого максимален. Под резервом времени вычислительного процесса на *j*-м ВМ будем понимать величину $T - \theta_j - w_{j0}$, где T — директивный момент времени завершения вычислений, θ_j — потребные затраты времени на реализацию на *j*-м ВМ вычислительного процесса, w_{j0} — момент времени его начала.

Если завершение вычислительного процесса, выполняемого на отказавшем ВМ, возможно на одном из исправных ВМ за директивно назначенное время, то отказоустойчивость ВС составит 1. Подобным образом определяется *d*-устойчивость ВС как способность завершить все вычислительные процессы за установленный срок при отказе *d* любых ВМ. Так как отмеченный показатель является в общем случае случайной величиной, зависящей от момента времени отказов ВМ и количества отказавших ВМ, возможно построить функцию $\psi_m(d)$, определяющую вероятность *m*-модульной ВС обеспечить своевременное завершение всех вычислительных процессов при отказе любых *d* ($d < m$) ВМ.

Определим вектор $\mathbf{Y}_{\langle m \rangle} = \langle \gamma_1, \ldots, \gamma_m \rangle$ моментов времени отказа 1, 2, …, *m* вычислительных модулей, до наступления которых ВС не сможет завершить все вычислительные процессы на оставшихся исправных ВМ, т. е. если одновременный отказ *d* вычислительных модулей произойдет в момент времени $\tau < \gamma_d$, то оставшиеся $m - d$ ВМ не смогут своевременно завершить все запланированные вычисления.

Значения элементов вектора $\mathbf{Y}_{\langle m \rangle}$ зависят от размещения контрольных точек и резервов времени вычислительных процессов на каждом ВМ.

Тогда функция $\psi_m(d)$ может быть определена следующим образом:

$$\psi_m(d) = \left(1 - F\left(\gamma_d\right)\right)\rho(d),$$

где $F(\tau)$ — функция распределения момента времени отказа ВМ, $\rho(d)$ — вероятность отказа ровно d ВМ.

Пример. Пусть момент времени отказа ВМ четырехмодульной ВС распределен равномерно на интервале [0, T], плотность вероятности одновременного отказа d ВМ задана биномиальным распределением $C_4^d \cdot 0{,}15^d \cdot 0{,}85^{4-d}$, а вектор

$$\mathrm{Y}_{\langle m \rangle} = \left\langle \frac{T}{12}, \frac{T}{10}, \frac{T}{8}, T \right\rangle.$$

Тогда функция $\psi_4(d)$ будет иметь значения, представленные в табл. 1.

Таблица 1

Пример функции $\psi_4(d)$

d	1	2	3	4
$\psi_4(d)$	0,3378	0,0878	0,0100	0,0000

Значение функции $\psi_m(d)$ при $d = m$ всегда нулевое, т. е. $\psi_m(m) = 0$, так как при отказе всех ВМ вычислительной системы завершить вычисления не представляется возможным.

Вычисление значений этой функции связано с планированием выполнения прерванных отказом ВМ вычислительных процессов на оставшиеся исправные ВМ.

Алгоритм имитационного моделирования функционирования многомодульной вычислительной системы с репликацией контрольными точками

Рассмотрим алгоритм имитационного моделирования функционирования многомодульной вычислительной системы с репликацией контрольными точками, с помощью которого исследовалась зависимость d-устойчивости ВС от вероятности одновременного отказа d ВМ на некотором интервале функционирования ВС, момент времени отказа на котором распределен равномерно.

Схема алгоритма имитационного моделирования изображена на рис. 3.

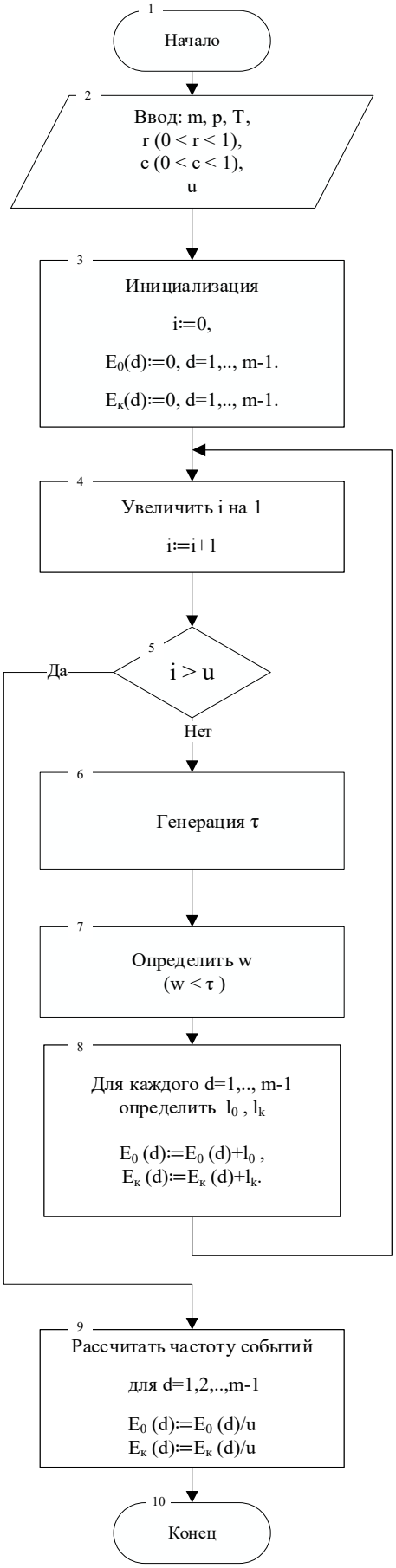

Рис. 3. Схема алгоритма имитационного моделирования

Таблица 2

Описание работы алгоритма имитационного моделирования

Шаг 1	Начало
Шаг 2	Ввод исходных данных: m — количество ВМ в ВС; p — вероятность события, приводящего к отказу ВМ; T — директивный срок завершения вычислений (правая граница интервала функционирования ВС); r — коэффициент резерва времени вычислений ($0 < r < 1$), определяющий долю времени от T, являющегося резервом вычислительного процесса (разница между директивным и реальным сроками завершения вычислительных процессов на каждом ВМ в штатном режиме); c — коэффициент затрат времени $0 < c < 1$, определяющий долю времени от $(1 - r)\,T$, которое тратится на создание контрольной точки; u — количество испытаний (прогонов) модели
Шаг 3	$i := 0$, $E(d) := 0, d = 1,.., m-1$
Шаг 4	$i := i + 1$
Шаг 5	Если $i > u$, то переход на шаг 9
Шаг 6	Генерация момента времени τ возникновения события, приводящего к отказу ВМ
Шаг 7	Определение момента времени w создания последней ($w < \tau$) контрольной точки
Шаг 8	Для всех значений $d = 1, 2, \ldots, m - 1$ определить возможность ($l_0 = 1$) или невозможность ($l_0 = 0$) завершить вычисления на оставшихся $m - d$ ВМ без создания КТ и соответствующие возможность ($l_к = 1$) или невозможность ($l_к = 0$) завершения вычислений с учетом создания КТ. Изменить значения массивов: $E_0(d) := E_0(d) + l_0$, $E_к(d) := E_к(d) + l_к$. Переход на шаг 4
Шаг 9	Рассчитать частоту событий, заключающихся в возможности завершения вычислений за директивное время для всех значений $d = 1, 2, \ldots, m - 1$: $E_0(d) := \frac{E_0(d)}{u}$, $E_к(d) := \frac{E_к(d)}{u}, \forall d = 1,.., m-1$
Шаг 10	Конец

Пошаговое описание работы алгоритма представлено в табл. 2.

В результате работы алгоритма массивы E_0 и $E_к$ будут содержать вероятностные значения h показателя d-устойчивости, $\forall d = 1,.., m-1$, для случаев неиспользования и использования КТ соответственно.

Показатель h показателя d-устойчивости ВС в результате работы приведенного алгоритма определяется как частота события, заключающегося в возможности завершения вычислений, начатых на всех ВМ, за установленное время при одновременном отказе d ВМ.

На рис. 3, *а* представлена зависимость вероятности h d-устойчивости восьмимодульной ВС от вероятности p события, приводящего к отказу ВМ, а на рис. 3, *б* — зависимость вероятности h d-устойчивости восьмимодульной ВС от резерва r времени вычислений.

Имитационное моделирование проводилось с учетом создания КТ. Фиксировались в первом случае резерв времени (r = 0,2), во втором — вероятность отказа ВМ (p = 0,5). Кроме того, в обоих случаях предполагалось, что временные затраты на создание каждой КТ не превышают 1 % времени, отводимого на вычисления.

На рис. 4 представлены те же зависимости, но только для прироста вероятности показателя d-устойчивости ВС при создании КТ по сравнению с ситуацией, в которой КТ не создаются.

Представленные зависимости показывают, что преимущества использования предложенного подхода с применением контрольных точек достигает 30 %.

Заключение

Совершенствование способов применения военной и специальной техники на основе вычислительных систем военного назначения в современных условиях требует дальнейшего исследования и новых подходов к организации их функционирования в условиях внешних возмущающих факторов [7, 8].

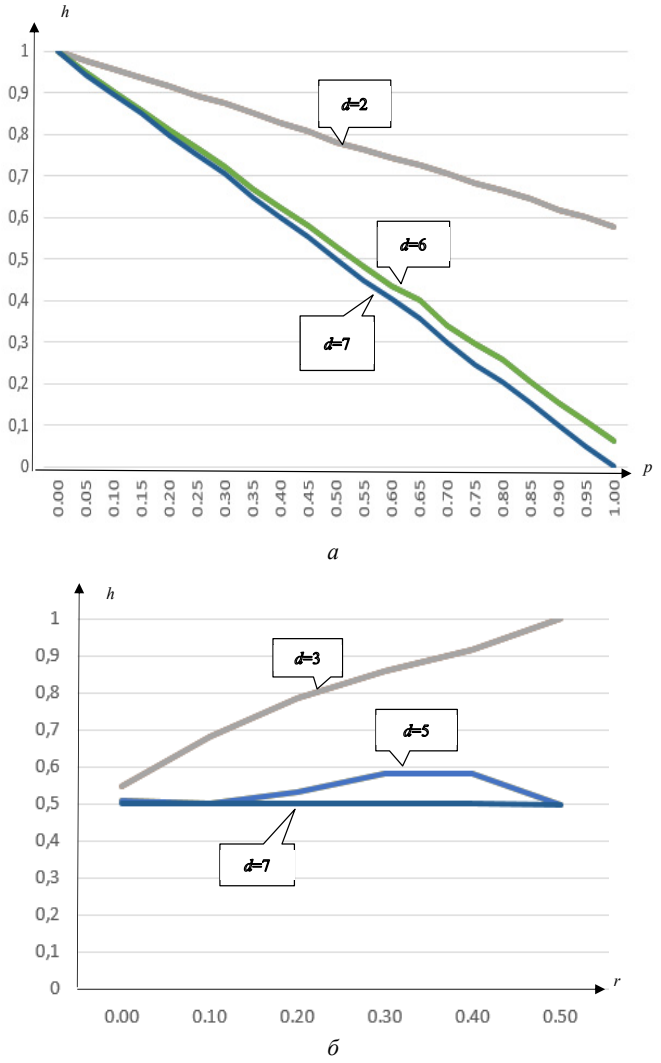

Рис. 3. Зависимости вероятности *d*-устойчивости вычислительной системы:
а — от вероятности события, приводящего к отказу ВМ;
б — от резерва времени вычислений

Жесткие требования по оперативности обработки целевой информации в режиме реального времени могут быть удовлетворены на основе метода «контрольных точек» с учетом структурной избыточности ВС и временной избыточности вычислительных процессов.

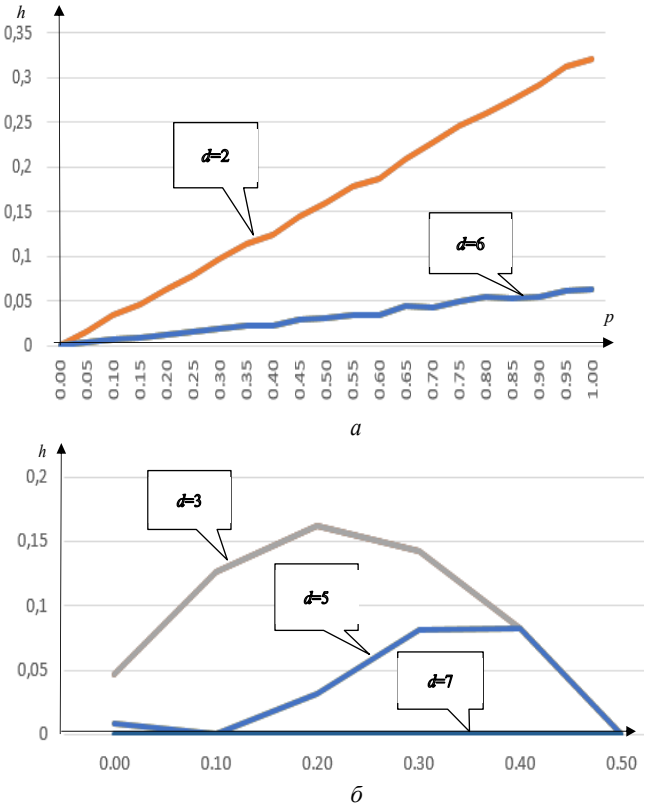

Рис. 4. Зависимости прироста *d*-устойчивости вычислительной системы:
а — от вероятности события, приводящего к отказу ВМ;
б — от резерва времени вычислений

Научная новизна представленной модели, в отличие от известных [9, 10], заключается в том, что она позволяет учитывать вид распределения момента времени отказов вычислительных модулей. Кроме того, хранение информации о КТ происходит на самих ВМ, а не на выделенном общем запоминающем устройстве.

Практическая значимость методики определяется возможностью ее применения для построения отказоустойчивых ВС, функционирующих в экстремальных ситуациях, что обеспечивает максимально возможную оперативность решения целевых задач.

СПИСОК ИСТОЧНИКОВ

1. Бондаренко А. А., Якобовский М. В. Обеспечение отказоустойчивости высокопроизводительных вычислений с помощью локальных контрольных точек // Вестник Южно-Уральского государственного университета. Серия «Вычислительная математика и информатика». 2014. Т. 3, № 3. С. 20–36.

2. Поляков А. Ю., Данекина А. А. Оптимизация времени создания и объема контрольных точек восстановления параллельных программ // Вестник СибГУТИ. 2010. № 2 (10). С. 87–100.

3. A Survey of Rollback-Recovery Protocols in Message-Passing Systems / E. N. Elnozahy, L. Alvisi, Y.-M. Wang, D. B. Johnson // ACM Computing Surveys. 2002. Vol. 34, Iss. 3. Pp. 375–408. DOI: 10.1145/568522.568525.

4. 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.

5. Зыкова С. С. Модель и алгоритм планирования параллельной обработки информации в отказоустойчивой бортовой вычислительной системе на основе временной избыточности вычислительного процесса // Интеллектуальные технологии на транспорте. 2023. № 4 (36). С. 28–33. DOI: 10.24412/2413-2527-2023-436-28-33.

6. ГОСТ Р ИСО/МЭК 25010—2015. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов = Information technology. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models: национальный стандарт Российской Федерации: утвержден и введен в действие приказом Федерального агентства по техническому регулированию и метрологии от 29 мая 2015 года № 464-ст: дата введения 2016-06-01. М.: Стандартинформ, 2015. 36 с.

7. Rathore N. Checkpointing: Fault Tolerance Mechanism // i-manager's Journal on Cloud Computing. 2017. Vol. 4, No. 1. Pp. 28–35. DOI: 10.26634/jcc.4.1.13756.

8. Koren I., Mani Krishna C. Fault-Tolerant Systems. Second Edition. Cambridge (MA): Morgan Kaufmann Publishers, 2020. 416 p.

9. Elnozahy E. N., Plank J. S. Checkpointing for Peta-Scale Systems: A Look into the Future of Practical Rollback-Recovery // IEEE Transactions on Dependable and Secure Computing. 2004. Vol. 1, Iss. 2. Pp. 97–108. DOI: 10.1109/TDSC.2004.15.

10. Optimal Checkpointing Period: Time vs. Energy / G. Aupy, A. Benoit, T. Hérault [et al.] // High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation (PMBS 2013): Revised Selected Papers of the 4th International Workshop (Denver, CO, USA, 18 November 2013). Lecture Notes in Computer Science. Vol. 8551. Cham: Springer International Publishing, 2013. Pp. 203–214. DOI: 10.1007/978-3-319-10214-6_10.

Дата поступления: 20.05.2025
Решение о публикации: 21.05.25

Failure Management and Fault Tolerance Techniques in a Multi-Module Computing System Based on Creation and Replication of Checkpoints

Denis A. Kochurov — Adjunct of the Department of Information and Computing Systems and Networks. Research interests: fault-tolerant computing systems. E-mail: deniskochurov45@yandex.ru

Mozhaisky Military Aerospace Academy, 13, Zhdanovskaya str., Saint Petersburg, 197198, Russia

For citation: Kochurov D. A. Failure Management and Fault Tolerance Techniques in a Multi-Module Computing System Based on Creation and Replication of Checkpoints. *Intellectual Technologies on Transport*2025, No. 2 (42), Pp. 103–112. DOI: 10.20295/2413-2527-2025-242-103-111. (In Russian)

Abstract. ***Introduction:*** *in order to enhance the efficiency of target information processing, it is necessary to adopt new approaches to the rapid detection and recovery from failures and faults to minimize the impact of such issues on the overall computing system.* ***Purpose:*** *to outline a technique for failure management and fault recovery in a multi-module computing system. This system implements periodic saving of calculations (checkpoints) and their exchange between all computing modules.* ***Results:*** *the problem of planning such a computing process has been outlined, including the determination of the optimal number and time points for creating checkpoints. The time points for creating*

checkpoints are determined based on the law of distribution of time points of computing module failures. ***Practical significance:*** *the results of the simulation modelling calculations conducted as part of the proposed approach demonstrate the feasibility of implementing the proposed technique.*

Keywords: *multi-module computing system, model of the computing process, checkpoint*

REFERENCES

1. Bondarenko A. A., Iakobovski M. V. Obespechenie otkazoustoychivosti vysokoproizvoditelnykh vychisleniy s pomoshchyu lokalnykh kontrolnykh tochek [Fault Tolerance for HPC by Using Local Checkpoints], *Vestnik Yuzhno-Uralskogo gosudarstvennogo universiteta. Seriya "Vychislitelnaya matematika i informatika" [Bulletin of the South Ural State University. Series "Computational Mathematics and Software Engineering"]*, 2014, Vol. 3, No. 3, Pp. 20–36. (In Russian)

2. Polyakov A. Yu., Danekina A. A. Optimizatsiya vremeni sozdaniya i obema kontrolnykh tochek vosstanovleniya parallelnykh programm [Optimization of Size and Creation Time of Parallel Programs Checkpoints], *Vestnik SibGUTI [The Herald of the Siberian State University of Telecommunications and Information Science]*, 2010, No. 2, Pp. 87–100. (In Russian)

3. Elnozahy E. N., Alvisi L., Wang Y.-M., Johnson D. B. A Survey of Rollback-Recovery Protocols in Message-Passing Systems, *ACM Computing Surveys*, 2002, Vol. 34, Iss. 3, Pp. 375–408. DOI: 10.1145/568522.568525.

4. Basyrov A. G., Zykova S. S., Koshel I. N., Kuznecov V. V. Metod otkazoustoychivoy parallelnoy obrabotki informatsii v bortovykh vychislitelnykh sistemakh letatelnykh apparatov na osnove vremennoy izbytochnosti vychislitelnogo protsessa [A Method of Fault-Tolerant Parallel Processing of Information in On-Board Computing Systems of Aircraft Based on the Temporary Redundancy of the Computing Process], *Aviakosmicheskoe priborostroenie [Aerospace Instrument-Making]*, 2023, No. 6, Pp. 33–39. DOI: 10.25791/aviakosmos.6.2023.1345. (In Russian)

5. Zykova S. S. Model i algoritm planirovaniya parallelnoy obrabotki informatsii v otkazoustoychivoy bortovoy vychislitelnoy sisteme na osnove vremennoy izbytochnosti vychislitelnogo protsessa [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], *Intellektualnye tekhnologii na transporte [Intellectual Technologies on Transport]*, 2023, No. 4 (36), Pp. 28–33. DOI: 10.24412/2413-2527-2023-436-28-33. (In Russian)

6. GOST R ISO/MEK 25010—2015. Informatsionnye tekhnologii. Sistemnaya i programmnaya inzheneriya. Trebovaniya i otsenka kachestva sistem i programmnogo obespecheniya (SQuaRE). Modeli kachestva sistem i programmnykh produktov [GOST R ISO/MEK 25010—2015. Information technology. Systems and software engineering. Systems and software Quality Requirements and Evaluation (SQuaRE). System and software quality models]. Effective from June 01, 2016. Moscow, StandartInform Publishing House, 2015, 36 p. (In Russian)

7. Rathore N. Checkpointing: Fault Tolerance Mechanism, *i-manager's Journal on Cloud Computing*, 2017, Vol. 4, No. 1, Pp. 28–35. DOI: 10.26634/jcc.4.1.13756.

8. Koren I., Mani Krishna C. Fault-Tolerant Systems. Second Edition. Cambridge (MA), Morgan Kaufmann Publishers, 2020, 416 p.

9. Elnozahy E. N., Plank J. S. Checkpointing for Peta-Scale Systems: A Look into the Future of Practical Rollback-Recovery, *IEEE Transactions on Dependable and Secure Computing*, 2004, Vol. 1, Iss. 2, Pp. 97–108. DOI: 10.1109/TDSC.2004.15.

10. Aupy G., Benoit A., Hérault T., et al. Optimal Checkpointing Period: Time vs. Energy, *High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation (PMBS 2013): Revised Selected Papers of the 4th International Workshop, Denver, CO, USA, November 18, 2013. Lecture Notes in Computer Science*, Vol. 8551. Cham, Springer International Publishing, 2013, Pp. 203–214. DOI: 10.1007/978-3-319-10214-6_10.

Received: 20.05.2025
Accepted: 21.05.2025