УДК 004.9

Катастрофоустойчивость бортовых вычислительных систем на базе микропроцессоров Эльбрус

Бочаров Никита Алексеевич[1], Елисеев Роман Владимирович[2],
Парамонов Николай Борисович[3], Янко Денис Викторович[4]

[1]АО «МЦСТ», 117105, Москва, ул. Нагатинская, д. 1, стр.23
[2]Управление военных представительств МО РФ
[3]ПАО Институт электронных управляющих машин им. И.С.Брука, 119334, Москва, ул. Вавилова, д. 24
[4]432 ВП МО, Москва, Россия
bocharov.na@phystech.edu, +7-916-734-64-37

Аннотация: Цель работы — разработка метода обеспечения катастрофоустойчивости бортовых вычислительных систем. Авторами были разработаны алгоритмы для повышения устойчивости бортовых систем управления к катастрофическим отказам и получены численные результаты, показывающие повышение надежности бортовых систем при использовании разработанных методов.

Ключевые слова: ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ, БОРТОВЫЕ СИСТЕМЫ УПРАВЛЕНИЯ, КАТАСТРОФОУСТОЙЧИВОСТЬ, НАДЕЖНОСТЬ, ГРУППОВОЕ УПРАВЛЕНИЕ РОБОТАМИ.

Disaster tolerance of on-board computer systems based on Elbrus microprocessors

Bocharov Nikita Alexeevich[1], Eliseev Roman Vladimirovich[2],
Paramonov Nikolay Borisovich[3], Yanko Denis Viktorovich[4]

[1]JSC «MCST», Moscow, Russia
[2]Department of military agency MO RF
[3] PJSC «Brook INEUM», Moscow, Russia
[4]432 VP MO, Moscow, Russia

Abstract: The purpose of this work is developing the method of disaster tolerance for on-board computer systems. The authors have developed algorithms for increasing the reliability of on-board control systems to catastrophic faults and numerical results of increasing the reliability of onboard systems using the developed methods are obtained.

Keywords: HIGH-PERFORMANCE COMPUTING SYSTEMS, ONBOARD CONTROL SYSTEMS, DISASTER TOLERANCE, RELIABILITY, GROUP CONTROL.

Введение

Современные наземные робототехнические комплексы (НРТК) могут эксплуатироваться в тяжелых условиях, а также в условиях боевых действий, которые могут провоцировать многочисленные отказы оборудования, причем как естественные отказы, так и преднамеренные, в следствии чего встает вопрос обеспечения катастрофоустойчивости системы управления робототехническим комплексом. Будем понимать под термином катастрофоустойчивость — способность к продолжению работы робота при возникновении внезапных катастрофических отказов через минимально короткий период времени. Существенным, но не решенным вопросом создания систем управления НРТК является оснащение вычислительной техникой, разработанной на базе отечественных микропроцессоров и программного обеспечения отечественной разработки[1].

Поскольку робототехника является одним из перспективных направлений применения вычислительных комплексов (ВК) и общего программного обеспечения (ОПО) семейства «Эльбрус»[2-4], то одной из целей данной работы было исследование применимости ВК и ОПО «Эльбрус»[5] для обеспечения катастрофоустойчивости бортовых вычислительных систем.

В статье введено определение катастрофоустойчивости бортовой вычислительной системы, проведен анализ показателей катастрофоустойчивости, проведен анализ способов обеспечения избыточности в НРТК, предложен алгоритм реконфигурации бортовой вычислительной сети с использованием ОПО и ВК «Эльбрус», предложены алгоритм распределенного хранения данных и алгоритм распределения нагрузки при отказе части оборудования.

Понятие катастрофоустойчивости

Разница между понятиями «отказоустойчивость» и «катастрофоустойчивость» заключаются в следующем. В понятии «отказоустойчивость» акцент делается на восстановление работоспособности после единичных, случайных, не связанных между собой отказов компонентов. Технология отработки таких отказов предполагает, как правило, что в работу вводятся резервные компоненты каждой подсистемы либо оставшиеся компоненты многократно дублированной подсистемы перераспределяют между собой работу независимо от того, что происходит в это время в других подсистемах.

В понятии «катастрофоустойчивость», примененном к бортовым вычислительным системам НРТК, главное — сохранение данных и продолжение работы в условиях массовых и, возможно, преднамеренных отказов систем и связанных между собой подсистем. Преднамеренные отказы могут вызваны сознательно нанесенным ущербом, таким как попадание снаряда в подсистему связи, или вывод из строя каких-либо узлов в результате столкновения. В качестве основного показателя в этом случае используется показатель доступности вычислительной системы, который характеризует степень возможности выполнения поставленной задачи в приемлемые сроки и с необходимым уровнем производительности. В данном контексте составляющими показателя доступности вычислительной системы являются показатели надежности аппаратно-программных средств вычислительной системы, а также показатель производительности данной системы, меняющийся в зависимости от схемы резервирования и количества отказов.

Показатели надежности

Бортовая вычислительная система по своей сути является системой с невосстанавливаемыми элементами, поскольку замена вышедших из строя элементов в автономно функционирующем роботе невозможна. Надежность таких систем определяется следующими показателями [ЛНТС]:

- вероятность безотказной работы P(t);
- оценка вероятности отказа Q(t);
- плотность распределения отказов f(t);
- интенсивность отказов $\lambda(t)$;
- средняя наработка на отказ $t_{ср}$.

Эти показатели определяется следующими выражениями:

$$P(t) = 1 - \frac{n(t)}{N} = 1 - Q(t);$$

$$f(t) = -\frac{dP(t)}{dt};$$

$$\lambda(t) = \frac{f(t)}{P(t)};$$

$$t_{\text{средн}} = \int_0^\infty t f(t) dt.$$

При решении задачи катастрофоустойчивости необходимо учитывать, что помимо основного (естественного) потока отказов, которые являются следствием ошибок, сбоев и т.д., есть поток отказов, вызванный целенаправленными попытками нанести повреждения роботу. Такие отказы могут быть, например, результатом выстрела в робота или тарана. Будем называть этот поток - потоком катастрофических отказов. В дальнейшем будем обозначать λ_1 – интенсивность потока естественных отказов, а λ_2 – интенсивность потока катастрофических отказов. Таким образом, общий поток отказов для бортового вычислительного комплекса будет определяться выражением:

$$\lambda = \lambda_1 + \lambda_2$$

В большинстве случаев задача обеспечения отказоустойчивости бортовых вычислительных комплекс уже решена, поэтому предлагаемый в данной статье метод сосредоточен на устойчивости к потоку катастрофических отказов. Также заметим, что при эксплуатации робота в условиях боевых действий или при иной угрозе нанесения целенаправленного вреда роботу интенсивность потока катастрофических отказов будет существенно выше чем интенсивность потока естественных отказов, т.е. $\lambda_2 \gg \lambda_1$. Таким образом, рассматриваемая задача обеспечения катастрофоустойчивости сводится к повышению вероятности безотказной работы при высокой интенсивности потока отказов.

При такой постановке задачи можем принять, что компоненты вычислительного комплекса имеют достаточно высокую надежность, чтобы обеспечивать работу робота при нормальных условиях, в которых возникают только естественные отказы. Поэтому предлагаемый метод обеспечения катастрофоустойчивости будет сосредоточен на обеспечении устойчивости к катастрофическим отказам, поток которых возникает кратковременно и интенсивно.

Режимы работы НРТК

Применительно к НРТК военного назначения (ВН) в период нормальной работы, будем считать, что НРТК ВН может функционировать в одном из трех режимов, характеризующихся согласно сложившейся обстановки и соответствующим потоком отказов:

- Режим подготовки. Роботы в составе группы движутся к назначенной цели, ведут активную работу с системами технического зрения, строят подробные карты проходимости и т.д. В этом режиме поток катастрофических отказов находится на практически нулевом уровне, т.е. $\lambda_2 \to 0$, а поток естественных отказов находится на обычном уровне. Риск получения физического урона или несанкционированного доступа в систему управления минимален. Обеспечение катастрофоустойчивости в таком режиме работы эквивалентно обеспечению отказоустойчивости, методы которого уже довольно широко изучены.
- Режим повышенной боевой готовности. Робот находится близко к зоне боевых действий и должен быть готов к переходу в режим боевых действий. В этом режиме уровень потока катастрофических отказов растет,

поэтому должны быть применены соответствующие методы, обеспечивающие своевременное переключение в режим боевых действий при возросшей угрозе возникновения катастрофических отказов. Риск получить физический урон минимален, риск несанкционированного доступа в систему управления возрастает, по сравнению с режимом подготовки.

- Режим боевых действий. Роботу непосредственно угрожает враждебные элементы, велик риск получения серьезного физического урона или несанкционированного доступа в систему управления. В данном режиме показатель потока катастрофических отказов λ_2 возрастает до своего максимального значения и становится много больше потока естественных отказов ($\lambda_2 \gg \lambda_1$).

Катастрофоустойчивость бортовой вычислительной системы

Использование одиночного вычислительного модуля (ВМ) в качестве бортовой вычислительной системы имеет очевидный недостаток. При работе робота в режиме подготовки, уровень потока отказов находится на низком уровне, и надежная работа бортовой управляющей системы НРТК ВН будет обеспечена на период примерно равный средней наработке на отказ. Однако, при резком росте потока отказов при переходе в режим повышенной боевой готовности и тем более режим боевых действий, обеспечение катастрофоустойчивости бортовой вычислительной системы НРТК ВН попросту невозможно. Любое внешнее воздействие может привести к полному отказу бортовой вычислительной системы в целом и, следовательно, к прекращению функционирования робота. Это обстоятельство является основной причиной необходимости использования разного рода избыточности при проектировании НРТК ВН.

Одним из основных методов обеспечения дополнительной надежности объекта является резервирование. Метод реализуется благодаря использованию дополнительных средств и возможностей, которые являются избыточными к минимально необходимым для выполнения требуемых функций. Наиболее частой реализацией метода резервирования является включение параллельно объекту резервирования дополнительных средств, которые полностью или частично дублируют его функции, и способны взять на себя его задачи при возникновении отказа.

В робототехнических комплексах применяются следующие основные виды резервирования:

- аппаратное (схемное, структурное);
- временное;
- информационное;
- функциональное;
- нагрузочное.

Одновременное применение двух и более видов резервирования является предпочтительным, так обеспечивает больший эффект в повышении надежности. Но основной вклад сохраняет за собой аппаратное резервирование, и оно должно реализовываться в первую очередь.

Аппаратное резервирование используется для резервирования всех критических компонентов бортовой вычислительной систем, начиная ВМ и заканчивая линиями связи. Информационное, функциональное и нагрузочное резервирование реализуется с помощью соответствующих алгоритмов, обеспечивающих работу бортовой вычислительной системы НРТК при отказе части оборудования. Эти алгоритмы рассмотрены ниже.

При реализованном резервировании отказ системы управления робототехнического комплекса в целом наступает только после отказа в основном вычислительном модуле и во всех резервных изделиях. Основным вычислительным модулем при этом считается тот модуль, который необходим для выполнения требуемых функций без использования резерва.

На практике различают следующие способы резервирования:

- резервирование по типу объекта
  - общее резервирование – объект резервируется целиком
  - раздельное резервирование – резервируются отдельные элементы объекта или их группы
- по кратности резервирования (кратность резерва k – отношение числа резервных элементов к числу резервируемых ими элементов)
  - резервирование с целой кратностью
  - резервирование с дробной кратностью
- по способу включения резервных элементов
  - постоянное резервирование
    - резервирование с нагруженным резервом - используется нагруженный резерв, т. е. находящийся в режиме работы основного элемента системы
    - резервирование с ненагруженным резервом – резерв содержит один или несколько резервных элементов, находящихся в ненагруженном режиме до начала выполнения ими функций основного элемента
    - резервирование с облегченным резервом – резервные элементы находятся в менее нагруженном режиме, чем основной элемент.
  - резервирование замещением - функции основного элемента передаются резервному только после отказа основного элемента

Для предварительной оценки преимуществ резервирования предположим, что при работе в режиме боевых действий вероятность отказа вычислительного модуля составляет 30%. Тогда вероятность безотказной работы нерезервированной системы будет равна 70%, а системы с одним резервным вычислительным модулем – 91%. Если добавить второй резервный вычислительный модуль, то тогда вероятность безотказной работы такой системы составит уже 97,3%. Из этого видим, что наличие даже одного резервного устройства повышает вероятность безотказной работы системы более чем на 20%.

На рисунке 1 приведены графики зависимости вероятности безотказной работы системы с резервированием замещением и при постоянном резервировании при различных кратностях резервирования n. Из графиков зависимостей видно, что общее резервирование замещением является более эффективным способом повышения уровня надежности системы, по сравнению с резервированием при постоянно включенном резерве. Его эффективность тем больше, чем больше $\lambda_0 t$, т. е., чем менее надежен резервируемый элемент системы. Однако, стоит заметить, что у постоянно включенного резерва есть свои преимущества, если резервируется вычислительный модуль целиком. Пока основной элемент не вышел из строя, можно использовать больше вычислительной мощности для решения бортовых задач и повышения эффективности функционирования робота.

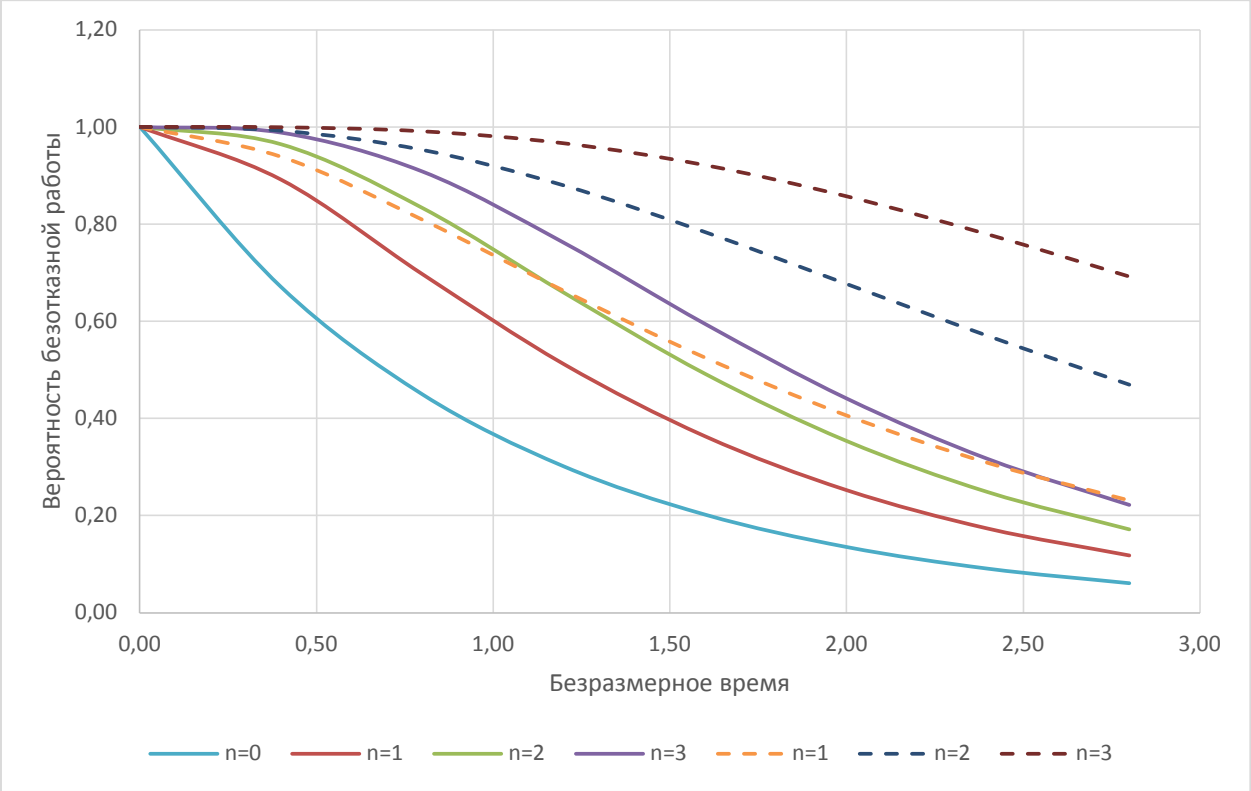

Рисунок 1 – Зависимость вероятности безотказной работы от безразмерного времени с резервированием замещением (пунктирные линии) и при постоянно включенном резерве (сплошные линии) для разных показателей кратности

Для резервирования разных компонентов бортовой вычислительной сети необходимо использовать разные способы резервирования. Например, не имеет смысла использовать раздельное резервирование для ВМ, поскольку это сильно усложнит его схему и конструкцию, и тем самым повысит вероятность его отказа, что скомпенсирует все преимущество, полученное при резервировании.

Алгоритм реконфигурации бортовой вычислительной сети при отказе части оборудования

При возникновении отказа в резервированной вычислительной системе, необходимо провести соответствующую реконфигурацию, которая будет учитывать сниженную производительность бортовой сети или отказ некоторых каналов связи. Вследствие вышесказанного, появилась необходимость разработать алгоритм реконфигурации бортовой вычислительной сети при отказе части оборудования.

Под бортовой вычислительной сетью будем понимать вычислительный многомашинный комплекс (ММК), включающий в себя множество машин одного типа (класса), объединённых в целях установления конфигурации, обмена служебными сообщениями операционной системы реального времени и сообщениями функциональных программ стандартными сетевыми средствами ЛВС, а также оптическими каналами типа БКЗ. Многомашинные комплексы, на основе которых реализуются системы реального времени, по существу являются специализированными локальными сетями, к которым предъявляется ряд жестких требований по составу физических каналов связи, по скорости передачи информации, по надежности и скорости доставки сообщений.

Концепцию архитектуры, резервируемой на основе однородных ВМ иллюстрирует рисунок 2, на котором показан пример бортовой вычислительной сети (ВС) и основные связи, реализующие одну из схем резервирования.

Каждый ВМ связан со всеми другими К контурами локальной сети. Каналы локальной сети в первую очередь предназначены для передачи информации управления многомашинным комплексом, диагностической и отладочной информации о работе системы реального времени.

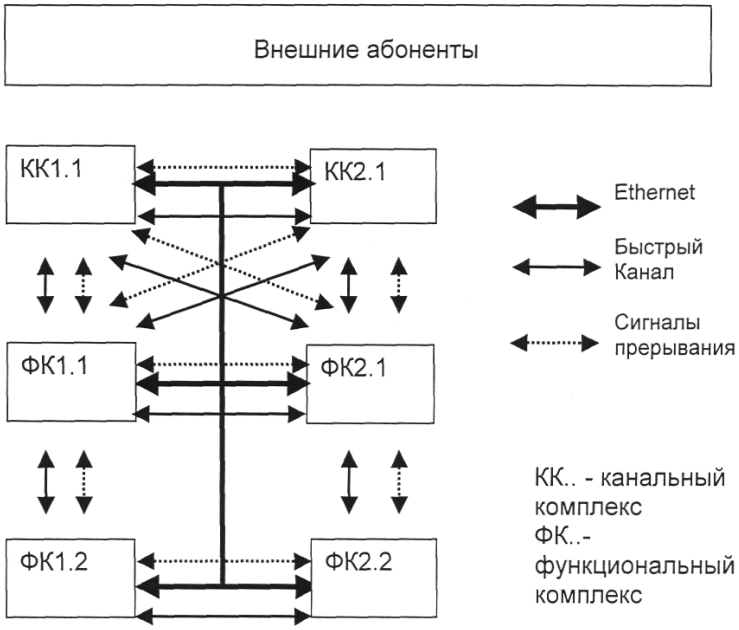

Рисунок 2 – ММК. Схема функциональных связей

При отказе одного из основных ВМ на резервном происходит рестарт соответствующего кода функциональных программ с последней согласованной контрольной точки. При этом ОСРВ выводит отказавший ВМ из работы, а резервному, принявшему на себя функции основного, присваивает статус основного.

Предполагается использование аппаратно-программных средства перезапуска системы при невыполнении ядром ОС контрольных действий по установки системного сторожевого таймера. Сторожевой таймер - это совокупность аппаратного обеспечения и программных средств, позволяющих перезапустить ОС вследствие программной или аппаратной ошибки.

В случае использования в качестве ОСРВ ОС «Эльбрус», процесс работы со сторожевым таймером выглядит следующим образом. Пользовательский процесс-демон через равные промежутки времени уведомляет драйвер сторожевого таймера через специальное устройство /dev/watchdog о том, что он до сих пор работоспособен. При получении такого уведомления драйвер программирует сторожевой таймер таким образом, чтобы отложить перезагрузку ОС на некоторое время. Если по какой-то причине уведомления не произошло, то сторожевой таймер перезагрузит ОС по истечению заданного интервала.

Перенос вычислений с основной машины на резервную может выполняться как автоматически (в случае отказа основной), так и оператором, восстановление же конфигурации после ремонта осуществляется только по командам оператора.

Возможность формирования требуемой конфигурации связей по определяется возможностями средств каскадирования модулей расширения шины S-bus.

К сети на основе ВМ с помощью стандартных сетевых средств типа Ethernet, а также посредством быстрых каналов могут подключаться машины других типов, использование которых позволит реализовать в конкретных СРВ разветвленные связи с различными устройствами.

Способом обеспечения конфигурирования ВС является применение программных средств для решения задач коммутации компонентов, описанных выше. Конфигурирование осуществляется посредством обмена служебными сообщениями по стандартной сети ЛВС. Исходная информация, необходимая для конфигурирования в процессе инициализации функциональных программ, формируется в файле конфигурации, описывающем МВК и пути коммутации между ними.

Инициализация ММК в целом осуществляется по следующей схеме. Сначала стандартными средствами общесистемного программного обеспечения запускается программа реального времени на главном ВМ, затем из этой программы путём обращения к соответствующей интерфейсной процедуре ОСРВ осуществляется инициализация и запуск программ реального времени на других ВМ. Исходные данные для инициализации должны содержаться в файле конфигурации.

В процессе инициализации на главной машине создается структура, описывающая конфигурацию сети, а также процесс-демон ОСРВ. Такие же структуры и процессы создаются на всех машинах ММК, связанных через стандартную сеть.

Кроме процессов-демонов ОСРВ на каждой машине для работы в реальном времени запускается программа реального времени. Программа реального времени на других машинах также обращается к этой

интерфейсной процедуре ОСРВ, которая предоставляет ей доступ к уже созданным структурам для взаимодействия в рамках ВС.

После успешной инициализации сети и запуска программ в режиме реального времени осуществляется взаимодействие ВМ на уровне функциональных программ. Таким образом, происходит синхронизация вычислительного процесса на уровне выполнения функциональных программ.

Организация взаимодействия «точка-точка» между ВМ при работе в реальном времени осуществляется путём использования интерфейсных процедур ОСРВ для синхронного и асинхронного ввода/вывода (записи, чтения, управления вводом/выводом и ожидания) в соответствии с логикой взаимодействия различных функциональных программ.

Распределенное хранение данных

Использование резервированного вычислительного комплекса ставит задачу распределенного хранения данных. Согласно [6-8] на данный момент надежных, формально и математически доказанных распределенных баз данных с равнозначными правами/функциями — не существует. Возможен только вариант с «выделенной» главной базой данных (master) и подчиненными (slave). Обмен информацией в такой архитектуре между базами данных агентов группы называется master-slave репликация [6].

Основная проблема использования распределенных баз данных – обеспечение консистентности (согласованность) данных [9]. На практике оказывается, что для достижения согласованности данных требуется большое количество служебных сообщений (например, чтобы убедиться, что имеющиеся у робота данные до сих пор актуальны), что в случае группы роботов и, соответственно, медленных и не всегда стабильных каналов связи, не позволяет добиться высокого уровня производительности. Среди рассмотренных в работе [9] для задачи обеспечения катастрофоустойчивости бортовых вычислительных систем идеально подходит модель консистентности по выходу. Данная модель не требует так много пересылок данных как модели строгой, последовательной или слабой консистентности, но, в то же время, минимизирует время обладания узлом уникальной информацией (что важно для обеспечения устойчивости системы ко сбоям отдельных узлов), так как в момент выхода из критической секции произведенные в общей памяти изменения должны быть распространены между остальными узлами сети.

Наиболее очевидное решение задачи управления распределёнными данными – выбрать один из ВМ так называемым сервером, и производить все операции через него (рис. 3). Например, когда модель консистентности требует от некоторого узла (скажем, №2) обновить у себя те или иные общие данные – обращаться за этими данными следует именно к серверу (узел №1). И наоборот – если модель требует от узла (например, №2) распространить его изменения общих данных по всем остальным узлам – узлу нужно делегировать эту задачу серверу (№1), передав ему и постановку задачи, и данные для неё. Сервер же самостоятельно свяжется с остальными узлами (№3 и №4) и передаст им нужную информацию. Обособление в распределённой системе какого-то узла потенциально опасно, так как выделенный узел становится слабым местом системы – выход из строя лишь одного данного узла может привести к тому, что вся система выйдет из строя. Воспрепятствовать этому можно, например, через резервирование узла-сервера (рис. 4).

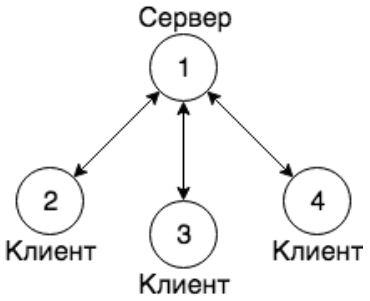

Рисунок 3 – Алгоритм с центральным сервером

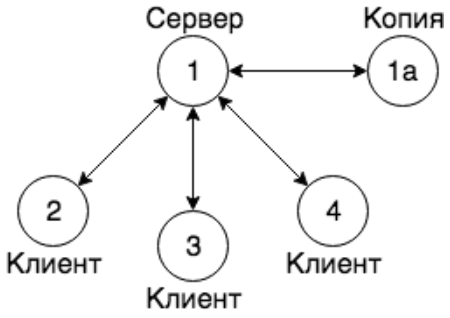

Рисунок 4 – Отказоустойчивый алгоритм с центральным сервером

Один из ВМ назначается «зеркалом» узла-сервера. И теперь, при выполнении любой операции записи сервер (узел №1) сначала информирует о ней свою копию (узел №1а), и только получив от копии ответ, продолжает свою работу. Если выйдет из строя узел-копия, узел-сервер это заметит по срабатыванию таймаутов на операции, с которыми сервер обращается к копии, и выделит другой узел в качестве своей копии.

Использование полной репликации позволит нескольким узлам одновременно не только читать, но и записывать одни и те же данные. Чтобы сохранять в таких условиях консистентность, все операции записи должны быть каким-то образом упорядочены. Для этого каждое своё оповещение сервер снабжает номером, который возрастает с каждой рассылкой. Благодаря данному номеру узлы могут быть уверены, что получают оповещения в правильном порядке (а пропустив какое-то из них, имеют возможность запросить его снова, указав в запросе соответствующий номер).

Так как данные распределены по всем узлам, выход из строя любого из них оказывается обратим без дополнительных усовершенствований алгоритма. Сервер может быть восстановлен, как и любой другой узел, так как всё, что ему нужно «знать», кроме состояния общей памяти, это последний использованный его предшественником номер.

Таким образом, при реализации описанных выше алгоритмов, обеспечивается сохранность данных при выходе из строя любого ВМ в НРТК. При необходимости алгоритм может быть масштабирован путем увеличения количества копий.

Распределение нагрузки при отказе части оборудования

В случае выхода из строя части резервированной управляющей системы робота, вычислительных ресурсов может перестать хватать для обеспечения выполнения его задач. В таком случае можно либо ограничить задачи робота необходимым минимумом (например, проложить дорогу до станции ремонта или оставаться на месте в роли транслятора потока данных между роботами в группе), либо задействовать свободные вычислительные ресурсы других ВМ и распределить задачи по ним.

Обобщенный алгоритм принятия решения о проведении реконфигурации состоит из следующих шагов:

1. Формируется булевый вектор, отражающий изменения в состоянии вычислительной сети:

$$R = \{^{1, S_i \neq S_{i-1}}_{0, S_i = S_{i-1}}\}$$

где S – множество исправных ВМ. Индексом i-1 обозначается предыдущая итерация выполнения алгоритма, а i – текущая. Единица означает, что по сравнению с предыдущей итерацией произошли изменения в состоянии вычислительной сети.

2. В зависимости от состояния вектора принимается решение об изменении состояния вычислительной сети и проведении реконфигурации согласно таблице реконфигурации или по некоторому критерию оптимальности.

3. Реализуется алгоритм выбранного на шаге 2 вида реконфигурации. Производится обнуление координат вектора R. В случае успешного проведения реконфигурации переходят к шагу 1. В случае неудачи – к шагу 4.

4. Реконфигурация невозможна по причине недостаточности аппаратных ресурсов или отказа функции с высоким уровнем критичности. Выдача сигнала неисправности робота. Конец функционирования робота.

Шаги 1-3 выполняются циклически на протяжении всего времени функционирования системы. Решение о выдаче сигнала неисправности может быть принято уже на шаге 1 при анализе входных данных об отказах. Например, при выходе из строя всех ВМ на одном роботе не имеет смысла осуществлять попытки реконфигурации. Для реализации этого подхода, помимо анализа вектора R, необходимо проверять заданные ограничения по количеству отказов. Кроме того, в перечень возможных состояний комплекса можно добавить такие варианты, при которых будет осуществляться проверка возможности выполнения функций с высоким уровнем критичности без оптимального их распределения. Это может оказаться полезным в сложных системах с большим количеством ВМ, в которых существует возможность такого распределения с целью сохранения ее работоспособности при множественных отказах.

Заключение

В данной статье рассмотрена катастрофоустойчивость бортовой вычислительной системы НРТК. Авторами было введено определение катастрофоустойчивости бортовой вычислительной системы, рассмотрены способы ее обеспечения с использованием избыточности разного типа. Были предложены соответствующие алгоритмы, которые учитывают особенности ВК и ОПО «Эльбрус».

Использование отечественных вычислительных средств и сертифицированного ОПО «Эльбрус» позволяет говорить о перспективах решения задач импортозамещения в области робототехники.

Авторы считают, что в данной работе новыми являются следующие положения и результаты: введено определение катастрофоустойчивости бортовой вычислительной системы НРТК, разработаны алгоритмы для обеспечения катастрофоустойчивости бортовой вычислительной системы НРТК с использованием ВК и ОПО серии «Эльбрус».

Благодарности

Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект №17-29-03297)

Литература

1. Бычков И.Н., Глухов В.И., Трушкин К.А. Доверенная программно-аппаратная платформа «Эльбрус». Отечественное решение для АСУ ТП КВО // Журнал «Информатизация и Системы Управления в Промышленности», № 1(49), 2014.- С.66-71.
2. Парамонов Н.Б., Ржевский Д.А., Перекатов В.И. Доверенная программно-аппаратная среда «Эльбрус» бортовых вычислительных средств робототехнических комплексов // Вопросы радиоэлектроники, сер. ЭВТ, 2015, №1.
3. Бочаров Н. А., Парамонов Н. Б., Тимофеев Г. С., Панова О. Ю. Производительность вычислительной техники с процессором «Эльбрус-8с» на задачах робототехнического комплекса // Наноиндустрия 2018. №82. С 79-84.
4. Бочаров Н.А., Парамонов Н.Б., Александров А.В., Славин О.А. Решение задач когнитивного управления группой роботов на многоядерных микропроцессорах «Эльбрус» // Труды II Международной научной конференции «Конвергентные когнитивно-информационные технологии» (Convergent'2017), Москва, 24-26 ноября, 2017. С 232-244.
5. Д.М. Альфонсо, Р.В. Деменко, А.С. Кожин, Е.С. Кожин, Р.Е. Колычев, В.О. Костенко, Н.Ю. Поляков, Е.В. Смирнова, Д.А. Смирнов, П.А. Смольянов, В.В. Тихорский. Микроархитектура восьмиядерного универсального микропроцессора «Эльбрус-8С» // Вопросы радиоэлектроники. 2016. Т. 4. № 3. С. 6–13.
6. Шилкин Е.А., Клопов И.Н., Остроухов А.В., Клеветов Д.В. Архитектура распределенных баз данных и ролей в реализации группового управления роботами // Современная наука и практика. 2017. №6-7 (23). С. 12-17.
7. Белоглазов Д.А., Гайдук А.Р., Косенко Е.Ю., Медведев М.Ю., Пшихопов В.Х. Групповое управление подвижными объектами в неопределенных средах // М.: ФИЗМАТЛИТ. 2015. 315 с.
8. Каляев И.А., Гайдук А.Р., КапустянС.Г. Модели и алгоритмы коллективного управления в группах роботов // М.: ФИЗМАТЛИТ. 2009. 280 с.
9. Бойко П.В. Система распределенной общей памяти для мультиагентных систем в IoT: диссертация кандидата технических наук. Санкт-Петербургский государственный университет, Санкт-Петербург, 2017.

References

1. Bychkov I.N., Glukhov V.I., Trushkin K.A. *Doverennaya programmno-apparatnaya platforma «Elbrus». Otechestvennoe reshenie dlya ASU TP KVO* // Informatizatsiya i Sistemy Upravleniya v Promyshlennosti. 2014. № 1(49). p.66-71. (in Russian).
2. Paramonov D., Rjevsky D., Perekatov V. *Trusted software and hardware environment"Elbrus" on-board computing means robotic complexes* // Voprosy radioelektroniki. 2015, №1. (in Russian).
3. Bocharov N. A., Paramonov N. B., Timofeev G. S., Panova O. Yu. *Performance of the computer systems with Elbrus-8S processor on the robotic systems tasks* // Nanoindustriya. 2018. №82. P. 79-84. (in Russian).
4. Bocharov N.A., Paramonov N.B., Aleksandrov A.V., Slavin O.A. *Solving of tasks of cognitive control a robots group in multi-core microprocessors «Elbrus»* // Selected Papers of the II International Scientific Conference "Convergent Cognitive Information Technologies" (Convergent 2017). Moscow, Russia, November 24-26, 2017. p. 232-244. (in Russian).
5. D. Alfonso, R. Demenko, A. Kozhin, E. Kozhin, R. Kolychev, V. Kostenko, N. Polyakov, E. Smirnova, D. Smirnov, P. Smolyanov, V. Tikhorskiy. *Eight-core «Elbrus-8C» processor microarchitecture* // Voprosy radioelektroniki. 2016. T. 4. № 3. p. 6–13. (in Russian).
6. Shilkin E.A., Klopov I.N., Ostroukhov A.V., Klevetov D.V. *Arkhitektura raspredelennykh baz dannykh i rolei v realizatsii gruppovogo upravleniya robotami* // Sovremennaya nauka i praktika. 2017. №6-7 (23). p. 12-17. (in Russian).
7. Beloglazov D.A., Gaĭduk A.R., Kosenko E.Yu., Medvedev M.Yu., Pshikhopov V.Kh. *Gruppovoe upravlenie podvizhnymi ob"ektami v neopredelennykh sredakh* // M.: FIZMATLIT. 2015. 315 p. (in Russian).
8. Kalyaev I.A., Gaĭduk A.R., KapustyanS.G. *Modeli i algoritmy kollektivnogo upravleniya v gruppakh robotov* // M.: FIZMATLIT. 2009. 280 p. (in Russian).
9. Boiko P.V. *Sistema raspredelennoi obshchei pamyati dlya mul'tiagentnykh sistem v IoT*: PhD thesis. St Petersburg University, St Petersburg, 2017. (in Russian).

Об авторах:

Бочаров Никита Алексеевич, магистр, инженер-программист 1 категории АО «МЦСТ», bocharov.na@phystech.edu

Елисеев Роман Владимирович, начальник службы управления военных представительств МО РФ

Парамонов Николай Борисович, доктор технических наук, профессор, главный научный сотрудник АО «МЦСТ», paramonov_n_b@mail.ru

Янко Денис Викторович, начальник группы, 432 ВП МО, janko_d@ineum.ru.

About authors:

Bocharov Nikita Alexeevich, Master, Engineer-programmer of the 1st category JSC «MCST», bocharov.na@phystech.edu

Eliseev Roman Vladimirovich, head of service of department of military agency MO RF

Paramonov Nikolay Borisovich, Doctor of Technical Sciences, Professor, Chief Researcher of PJSC «Brook INEUM», paramonov_n_b@mail.ru

Yanko Denis Viktorovich, head of group, 432 VP MO, janko_d@ineum.ru