Э. М. Мамедли, Н. А. Соболев, Механизмы операционных систем, обеспечивающие отказоустойчивость в управляющих многомашинных вычислительных системах, *Автомат. и телемех.*, 1995, выпуск 8, 3–63

УДК 681.324

© 1995 г. Э. М. МАМЕДЛИ, канд. техн. наук,
Н. А. СОБОЛЕВ
(Институт проблем управления РАН, Москва)

МЕХАНИЗМЫ ОПЕРАЦИОННЫХ СИСТЕМ, ОБЕСПЕЧИВАЮЩИЕ ОТКАЗОУСТОЙЧИВОСТЬ В УПРАВЛЯЮЩИХ МНОГОМАШИННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Формулируются закономерности, связывающие способы управления и восстановления вычислительного процесса управляющих отказоустойчивых многомашинных вычислительных систем. Определяется влияние характера (детерминированного или случайного) распределения ресурсов вычислительной системы и реализации синхронного режима взаимодействия между ЦВМ, выполняющими копии прикладных задач, на качество отказоустойчивости. Показывается двойственный характер проектирования эффективных средств управления вычислительным процессом в системах с детерминированным и случайным распределением ресурсов. Устанавливается, что попытки реализовать универсальную операционную систему, настраиваемую на любую конкретную среду реального времени, неизбежно ведут к снижению отказоустойчивости.

1. Введение

В многомашинных вычислительных системах (МВС) жесткого реального времени (РВ) отказоустойчивость обеспечивается с помощью резервирования аппаратуры – ЦВМ и шин, резервирования (копирования) системных и прикладных задач, а также использования дополнительных аппаратурных и программных средств, управляющих восстановлением вычислительного процесса (ВП) после проявления физических неисправностей, проектных и интерактивных ошибок. Именно эти дополнительные средства осуществляют контроль идентичности результатов выполнения копий прикладных задач в резервированных ЦВМ, обнаружение рассогласований, вызванных неисправностями и ошибками, и восстановление ВП. Ограниченные ресурсы вынуждают реализовать восстановление по фрагментам, чередуя их с решением прикладных задач. При этом, естественно, требуется эффективно организовывать работу средств обеспечения отказоустойчивости в РВ, т.е. обеспечить гарантированное решение прикладных задач с заданным уровнем отказоустойчивости при минимальных дополнительных затратах ресурсов системы.

В обзорных публикациях до начала 80-х годов средства обеспечения отказоустойчивости МВС рассматривались независимо от управления ВП [1 – 6]. С появлением МВС, выполняющих вычисления в нескольких потоках команд, и аппаратурно-программных средств обеспечения отказоустойчивости возникла необходимость организовывать восстановление ВП в РВ с учетом свойств как структуры МВС, так и элементов операционной системы (ОС), управляющих параллельными вычислениями. Тем не менее, в обзорных работах [7 – 9] попытки объяснить с системных позиций свойства средств обеспечения отказоустойчивости проводились по упрощенной схеме. Вводилась некоторая классификация МВС по какому-либо признаку, действительно имевшему существенное влияние на отказоустойчивость. Далее утверждалось, что указанные признаки полностью определяют способы восста-

новления. Затем приводились примеры конкретных систем, подтверждающие зависимость способа восстановления от признака, по которому была проведена классификация.

Во всех этих обзорах продемонстрированы общие свойства средств обеспечения отказоустойчивости в соответствии с выбранной классификацией. В то же время многие закономерности их работы в зависимости от управления ВП остались скрытыми. Ключевой недостаток указанных обзоров – нечеткий выбор признаков, по которым проводилась классификация. При этом часть факторов, существенно влияющих на восстановление, выпадала из поля зрения исследователя, что приводило к искусственному объединению принципиально различных систем в один и тот же класс и, как следствие, к ошибочным выводам. Так, например, в [7, 9] в один и тот же класс МВС входят как управляющие, так и информационные системы, отказоустойчивость которых обеспечивается принципиально различными способами, причем для первых они имеют более развитый характер. Таким образом, нечеткость классификации отказоустойчивых МВС жестко определяет низкое качество исследований. В результате этих ошибок в наибольшей степени "пострадали", естественно, управляющие системы, которые в отличие от информационных не могут иметь единой концептуальной модели, объясняющей способ восстановления.

Действительно, информационным системам присуща общая концепция предоставления ресурсов памяти и процессорного времени резервированным копиям и организации их взаимодействия. В них программы и обрабатываемые ими данные могут быть разделены и хранятся в памяти разных ЦВМ. При этом требуется обеспечить целостность работы с копиями данных разными программами. Изменение значений данных любой программой должно синхронно воспроизводиться во всех копиях. Именно это свойство позволяет построить глобальную модель синхронизации и единообразно анализировать способы восстановления [10]. В управляющих же системах каждая программа и обрабатываемые ею данные хранятся в памяти одной и той же ЦВМ, организация работы с копиями определяется способами управления параллельными вычислениями, а их несколько. Поэтому для управляющих систем нет единой модели, объясняющей все способы восстановления.

Таким образом, во всех существующих обзорах в качестве признака, по которому проводилась классификация отказоустойчивых МВС, выбирались элементы структуры (общая шина, общая память, степень связности, частные свойства механизмов управления параллельными процессами) и не уделялось должного внимания ОС. В действительности же только способ управления ВП определяет способ его восстановления после проявления физических неисправностей, проектных и интерактивных ошибок. Анализ многообразия способов управления ВП с точки зрения обеспечения отказоустойчивости управляющих МВС никем не проводился. Этому посвящена данная работа.

2. Некоторые определения и постановка задачи

Все определения в обзоре выделены курсивным шрифтом. Они должны содержать ключевые свойства структуры управляющих отказоустойчивых МВС, способов управления и восстановления ВП, а также функций элементов ОС, совместно реализующих их. Постановка задачи должна содержать ограничения, определяющие рассматриваемые ОС, а также метод исследования, с помощью которого будут определены закономерности, связывающие способы управления и восстановления ВП. ВП рассматривается на уровне прикладных задач.

Управляющие и информационные системы различаем по признаку размещения программы и данных. *В управляющих МВС при выполнении любой прикладной задачи взаимодействие между программой и ее данными осуществляется в памяти*

одной и той же ЦВМ. В информационных системах имеется хотя бы одна прикладная задача, программа и данные которой размещены в памяти разных ЦВМ. В дальнейшем речь идет в основном только об управляющих МВС.

В [11] впервые определена *отказоустойчивость, как свойство архитектуры ВС, которое позволяет ей, как логической машине, продолжать работу тогда, когда в реальной системе, являющейся ее носителем, возникают разнообразные отказы и сбои компонентов.* Любая реализация отказоустойчивости основана на периодической обработке средствами ОС контрольных точек (КТ), априорно введенных в прикладные задачи. *В КТ прерывается (или заканчивается) выполнение прикладной задачи, после чего между ЦВМ происходит: обмен текущими результатами выполнения задачи (исходными данными, принимаемыми извне); определение признака возможного несовпадения (недопустимого отклонения) значений обрабатываемых данных; при наличии признака – восстановление ВП. Последовательность действий (шагов), выполняемых ОС каждой ЦВМ, участвующей в обработке КТ, назовем процедурой обработки КТ.* Шаги процедуры обработки КТ различаются в зависимости от способа управления выполнением прикладных задач в МВС РВ.

Под ВП понимается реализация функций элементов ОС, которые в РВ одновременно обеспечивают управление: решением прикладных задач в директивные сроки и обеспечением отказоустойчивости (обработкой КТ). Управление выполнением прикладных задач и обработкой КТ в любой отказоустойчивой МВС РВ осуществляется с помощью внешних и внутренних событий. *Под внешними событиями имеется в виду поступление исходных данных из внешней среды, необходимых либо для начала, либо для завершения выполнения одной или нескольких прикладных задач. Под внутренними событиями понимается изменение состояния МВС, вызванное возникновением физических неисправностей, проявлением проектных или интерактивных ошибок. Эти события совместно с моментами начала и окончания решения системных и прикладных задач полностью определяют среду РВ в рассматриваемых МВС.* По признаку постоянного или изменяющегося распределения ресурсов между прикладными задачами будем различать стационарный и нестационарный ВП. *Стационарный ВП выполняется под управлением внешних событий при постоянном распределении ресурсов. Выполнение нестационарного ВП может быть вызвано внутренними событиями и реализует изменение распределения ресурсов между задачами. Иначе, стационарный процесс соответствует только выполнению прикладных задач, нестационарный – одновременному выполнению прикладных задач и восстановлению стационарности ВП (обработке КТ).*

Будем различать моменты наступления внутреннего события (возникновения неисправности), начала восстановления (проявления (обнаружения) неисправности в процедуре обработки КТ, ближайшей к моменту ее возникновения) и его окончания. От возникновения до проявления неисправность имеет скрытый от средств обеспечения отказоустойчивости характер. Момент окончания восстановления совпадает с завершением обработки КТ, в которой неисправная часть системы однозначно определена всеми исправными ЦВМ. В зависимости от способа восстановления, реализованного в МВС, его окончание может наступить по завершении одной или s процедур обработки КТ с момента проявления неисправности (рис. 1).

Любая отказоустойчивая МВС состоит из комплексов ЦВМ. *Группу ЦВМ, выполняющих копии одной и той же задачи, будем называть комплексом. Система, состоящая из нескольких комплексов, – это отказоустойчивая МВС, в которой разными комплексами одновременно выполняются несколько прикладных задач.* Способы реализации ВП в любой управляющей МВС необходимо рассматривать на уровне комплексов.

2.1. *Структура МВС, способы реализации ВП.* В классификации МВС, введенной в [12], за основу взят параллелизм элементарных операций, реализованный с

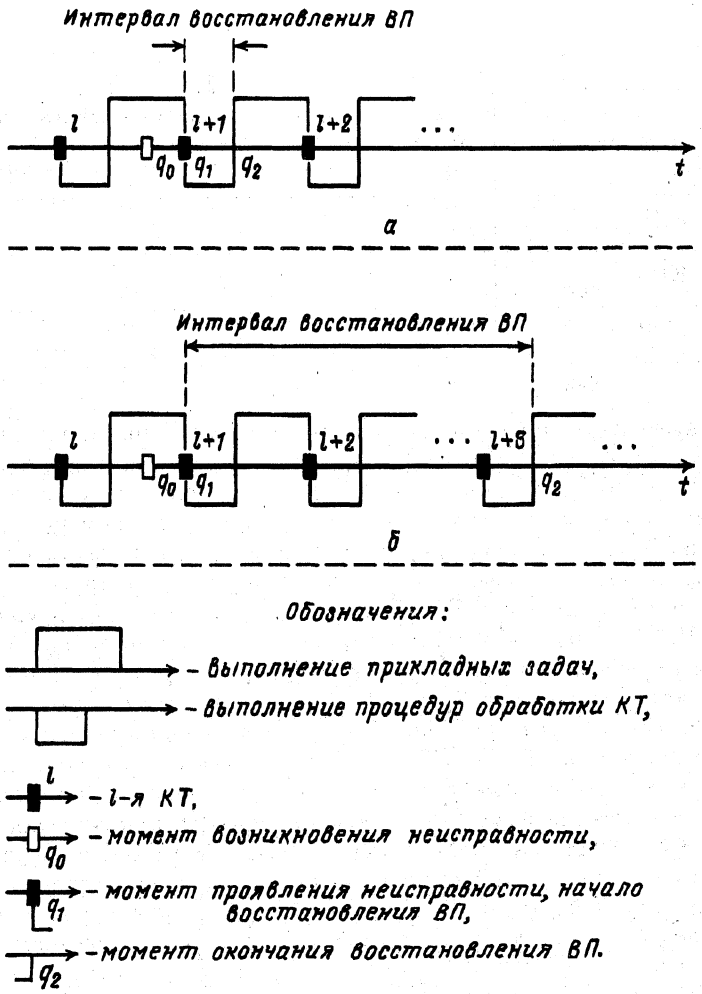

Рис. 1. Моменты возникновения и проявления неисправностей, интервалы обработки КТ и восстановления ВП. *а* – восстановление ВП с помощью одной процедуры обработки КТ; *б* – восстановление ВП с помощью *s* процедур обработки КТ

использованием аппаратурной поддержки. Он формирует четыре варианта на основе сочетаний двух элементов – структуры управления и структуры потока обрабатываемых данных, которые могут быть либо одиночными, либо множественными. Тем самым были определены следующие варианты архитектуры МВС: с одиночными потоками команд и данных (ОКОД); с одиночным потоком команд и множественным потоком данных (ОКМД); с множественным потоком команд и одиночным потоком данных (МКОД); с множественными потоками команд и данных (МКМД) – в строгом смысле не реализованный вариант параллелизма элементарных операций. Аппаратурно-программные средства обеспечения отказоустойчивости, являющиеся предметом исследования в обзоре, реализованы только в тех МВС, структура которых образована из одного или нескольких (*multiple*) комплексов типа ОКОД. Именно для них характерна связь элементов ОС, осуществляющих управление и восстановление ВП.

Таблица 1

Режимы внутрикомплексного обмена

Варианты комплексов	Режим внутрикомплексного обмена	
	синхронный	асинхронный
КПР	+	+
КРЗ	+	–
КСР	+	–

Существует много определений многомашинных ВС, причем под многомашинными понимают системы, как обрабатывающие несколько потоков команд и данных, так и использующие резервирование ЦВМ при выполнении одного потока команд. Использование одинаковых терминов в обоих случаях правомерно. Для того чтобы облегчить изложение, условно разделим причины использования многомашинных ВС: в одном случае для повышения отказоустойчивости, в другом – производительности. *Под многомашинной ВС, используемой только для резервирования вычислений одной ЦВМ, подразумевается комплекс. Под многомашинной ВС, используемой только для повышения производительности, подразумевается система из нескольких нерезервированных ЦВМ. Одновременное использование многомашинной ВС для повышения отказоустойчивости и производительности – это система, состоящая из нескольких комплексов.* Таким образом, понятие комплекса вводится только для того, чтобы различать характер использования многомашинных ВС.

2.1.1. *Способы реализации ВП в комплексе.* Способы управления ВП в комплексе различаются по характеру участия каждой исправной ЦВМ во внутрикомплексном обмене, а именно: активное/пассивное использование собственного результата выполнения прикладной задачи и синхронный/асинхронный режим обмена. *Копию задачи назовем активной, если результат вычислений передается в КТ во все остальные исправные ЦВМ в комплексе. В противном случае копию задачи назовем пассивной.* ЦВМ может выполнять либо активную, либо пассивную копию, причем в любом комплексе по крайней мере одна копия является активной. В обмене исходными данными участвуют все исправные ЦВМ комплекса, тогда как в обмене результатами вычислений – только те из них, которые выполняют активные копии. В стационарном ВП вычисление по пассивным копиям может быть блокировано. *В зависимости от соотношения числа ЦВМ, используемых для работы с активными или пассивными копиями задачи, будем различать следующие варианты комплекса: с постоянным резервированием (КПР, parallel redundancy), резервированный замещением (КРЗ, standby redundancy) и со смешанным резервированием (КСР)* (табл. 1).

Пусть N – общее число ЦВМ в комплексе, из них n_1 работают с активными, n_2 – с пассивными копиями. В любом варианте $N = n_1 + n_2$, $N \geqslant 2$, $n_1 \geqslant 1$. *В КПР все N ЦВМ являются основными и выполняют активные копии* ($n_1 \geqslant 2$, $n_2 = 0$). *КРЗ состоит из одной основной (выполняется активная копия) и нескольких резервных ЦВМ (выполняются пассивные копии, $n_1 = 1$, $n_2 \geqslant 1$). В случае отказа основной ЦВМ одна из резервных, определяемая по некоторому правилу, принимает на себя ее функции. КСР состоит из не менее двух основных и по крайней мере одной резервной ЦВМ* ($n_1 \geqslant 2$, $n_2 \geqslant 1$). Во всех известных из [13 – 20] КСР не менее трех ЦВМ были основными. В восстановлении ВП после сбоя участвовали только они. После отказа одной из основных ЦВМ ее замещение осуществлялось с использованием копирования памяти в одной из резервных, которую требовалось

перевести в основные. Та же самая процедура копирования памяти применима в КПР для восстановления после сбоев с нарушением содержания области хранения программ и констант в оперативной памяти неисправной ЦВМ. Поэтому в дальнейшем не будем рассматривать восстановление в КСР, а ограничимся рассмотрением этой процедуры в КПР.

Режим внутрикомплексного обмена данными будем называть синхронным, если при выполнении каждой такой операции ВП в передающей ЦВМ приостанавливается до получения от принимающей сигнала успешного/безуспешного приема. Режим внутрикомплексного обмена назовем асинхронным, если операции обмена осуществляются без получения сигнала подтверждения приема. Комплекс ЦВМ, в котором внутрикомплексный обмен выполняется в синхронном (асинхронном) режиме, будем называть синхронным (асинхронным). В синхронном КПР, использующем единую службу системного времени для синхронизации всех ЦВМ, копии одной и той же задачи выполняются одновременно с одинаковыми значениями данных, тогда как в асинхронном КПР неодновременно и с различающимися на допустимую величину значениями. Допустимые пределы расхождения во времени выполнения активных копий в асинхронном КПР не превышают периода выполнения прикладной задачи. Таким образом, КПР может быть либо синхронным, либо асинхронным. КРЗ и КСР могут быть только синхронными, поскольку при обработке КТ все резервные ЦВМ должны передавать сигнал подтверждения приема от основных. Он является единственным источником информации о состоянии каждой резервной ЦВМ, которая должна быть передана всем остальным в комплексе.

Способы восстановления в комплексе различаются, главным образом, в зависимости от того, выполняются исправными ЦВМ относительно неисправной восстановление после сбоя (блокирование после отказа) или нет. *Восстановление без вмешательства в работу неисправной ЦВМ называется стратегией маскирования.* Эта стратегия реализуема только в КПР при условии, что число исправных ЦВМ превышает число неисправных. В дальнейшем не будет рассматриваться реализация маскирования, поскольку она не зависит от управления ВП в комплексе и в МВС. *Восстановление с вмешательством исправных ЦВМ в работу неисправной называется стратегией локализации.* Эта стратегия применима как в КПР, так и в КРЗ. Реализующая ее процедура обработки КТ выполняется при взаимодействии элементов ОС, управляющих ВП и обеспечивающих его восстановление. Понятно, что именно эта стратегия должна быть исследована для определения влияния управления ВП на отказоустойчивость.

2.1.2. *Способы реализации ВП в МВС, состоящей из нескольких комплексов.* В таких системах для любого момента времени можно указать: число одновременно работающих комплексов, распределение прикладных задач и ЦВМ между ними. Режим межкомплексного обмена может быть только синхронным, что вызвано необходимостью информировать передающую и принимающую ЦВМ о его успешном/безуспешном завершении. Способы управления ВП определяются тем, как реализованы в ОС функции распределения ресурсов при формировании комплексов и синхронного режима межмашинного обмена.

Ключевым признаком управления ВП, кардинальным образом влияющим на способы его восстановления, является характер (детерминированный или случайный) распределения ресурсов между прикладными задачами. *В МВС с детерминированным распределением ресурсов расписание выполнения прикладных задач, обработки КТ и межмашинного обмена составлено заранее и используется как для управления ВП, так и для его восстановления. ОС таких систем реализует единую службу системного времени для всех ЦВМ, с помощью которой (согласно расписанию) осуществляется диспетчеризация прикладных задач и межмашинного обмена. В МВС со случайным распределением ресурсов априорное расписание выполнения прикладных задач отсутствует, каждая ЦВМ имеет независимую службу времени,*

а диспетчеризация задач и межмашинного обмена выполняется согласно заранее определенным условиям (событиям), при которых задача переходит из пассивного в активное состояние. В ОС таких систем реализованы механизмы, управляющие перемещением прикладных задач между ЦВМ и разрешающие конфликты при попытке одновременного захвата общих ресурсов (ЦВМ или шин) разными задачами. МВС, в которой для любых интервалов времени работы принадлежность каждой ЦВМ определенному комплексу не изменяется, будем называть статической. Систему, в которой одна и та же ЦВМ в зависимости от выполняемой ею задачи принадлежит разным комплексам, назовем динамической. Любую систему со случайным распределением ресурсов следует понимать как динамическую, тогда как МВС с детерминированным распределением ресурсов изначально проектируется либо как статическая, либо как динамическая, что проявляется в структуре межмашинных связей. В структуре статической МВС комплексы ЦВМ представлены явным образом, поскольку для внутри- и межкомплексного обменов используются разные группы шин. В структуре же динамической МВС распределение ЦВМ между комплексами является скрытым благодаря тому, что связи каждой ЦВМ со всеми остальными организованы одинаковым образом.

Синхронный режим межмашинного обмена может быть реализован различными способами в зависимости от структуры управления обменом (централизованной или распределенной) и от возможного использования в ОС исполнительной системы параллельного языка высокого уровня (ЯВУ). В системах с детерминированным распределением ресурсов используется распределенная структура управления межмашинным обменом с помощью единой службы системного времени. Поэтому специфика других способов реализации синхронного режима обмена проявляется только в МВС со случайным распределением ресурсов. *В системе с централизованной структурой управления обменом одна фиксированная ЦВМ является обязательным посредником при выполнении всех операций обмена между любыми двумя из остальных ЦВМ. В системе с распределенной структурой управления каждая ЦВМ выполняет обмен с остальными без посредников.* В ОС для управления параллельными задачами, в том числе для реализации синхронного режима межмашинного обмена, может быть использована либо собственная системная программа, либо исполнительная система параллельного ЯВУ. Языковая реализация управления параллельными процессами позволяет эффективно использовать процессорное время, если ОС и прикладные задачи используют один и тот же ЯВУ. Вместе с тем, использование языковых механизмов для реализации синхронного режима межмашинного обмена вынуждает соблюдать ограничения при управлении ВП.

В любой статической МВС восстановление реализовано только на уровне комплексов. В динамических системах восстановление после сбоев выполняется также на уровне комплексов, тогда как после отказов выполняется дополнительная функция – изменение распределения прикладных задач между исправными ЦВМ. Способы ее выполнения полностью определяются механизмами формирования комплексов в этих системах.

2.2. *Элементы ОС отказоустойчивых МВС. Совокупность программ, выполняющих любую конкретную функцию реализации ВП, назовем элементом ОС. Элементы ОС, функции которых определяют способы реализации ВП, назовем основными, остальные – вспомогательными.* Основными элементами ОС отказоустойчивых МВС являются супервизор, ядро и гипервизор, вспомогательными – средства межмашинного обмена и встроенного контроля ЦВМ. *Каждый элемент ОС выполняет одну или несколько функций. Функции могут быть: простыми (неделимыми с точки зрения анализа их реализации); сложными упорядоченными (состоящими из нескольких простых, выполняемых всегда в определенной последовательности); сложными неупорядоченными (состоящими из нескольких простых, выполняемых в зависимости от сложной упорядоченной). Будем называть конкретное выпол-*

Таблица 2

Функции элементов ОС, реализующие управление ВП

Элементы ОС	Функции элементов ОС, выполняемые в стационарном и нестационарном процессах
Супервизор	Диспетчеризация прикладных, системных задач, обмена данными с другими ЦВМ и с абонентами во внешней среде МВС
Ядро	Реализация синхронного режима выполнения межмашинного обмена. Управление ресурсами комплекса, корректирующее работу супервизора (только в нестационарном процессе)
Средства межмашинного обмена	Передача данных между задачами, выполняемыми разными ЦВМ (комплексами)

нение простой функции механизмом, сложной упорядоченной – процедурой (последовательностью механизмов). Таким образом, способы реализации ВП по сути представляют собой совокупность механизмов и процедур. Определим функции каждого элемента ОС (табл. 2 и 3).

Ядро выполняет две функции: реализует синхронный режим межмашинного обмена; управляет ресурсами комплекса при восстановлении ВП. В нерезервированных МВС синхронный и асинхронный режимы реализуют средства межмашинного обмена, работающие независимо от остальных элементов ОС. В отказоустойчивых МВС функция синхронизации одновременно связана с управлением и восстановлением ВП, поэтому она должна выполняться ядром во взаимодействии с другими основными элементами ОС (супервизором и гипервизором). Способы реализации асинхронного режима обмена в отказоустойчивых и нерезервированных МВС совпадают. *Под управлением ресурсами комплекса понимается искусственная коррекция работы супервизора и средств внутрикомплексного обмена, позволяющая в течение нескольких КТ после момента обнаружения неисправности иметь в каждой ЦВМ достаточную информацию о признаках ее проявления для того, чтобы неисправный элемент системы был ими однозначно определен. В число действий, реализуемых при такой коррекции, входят запоминание/восстановление в памяти значений параметров состояния ВП, обеспечивающих возможность восстановления с помощью процедуры рестарта; перераспределение полномочий управления шинами между ЦВМ; искусственная генерация признаков безуспешного завершения внутрикомплексного обмена с ЦВМ, которую часть исправных ЦВМ однозначно определили, как неисправную. Под процедурой рестарта понимается способ восстановления ВП в комплексе ЦВМ, требующий повторного выполнения прикладных задач с исходными данными, записанными в КТ, обработанной до момента обнаружения неисправности.* Именно ядро осуществляет связь средств управления и восстановления ВП, так что только в нем надо искать причины различий способов реализации отказоустойчивости в управляющих системах с различными способами управления ВП.

Гипервизор осуществляет выбор согласованного значения результата выполнения активных копий задачи, определяет неисправную ЦВМ (шину) и осуществляет восстановление ВП. Таким образом, функции гипервизора состоят в реализации каждого шага процедуры обработки КТ в каждой ЦВМ, при поддержке ядра и средств межмашинного обмена. В публикациях, посвященных ана-

Таблица 3

Функции элементов ОС, реализующие восстановление ВП

Элементы ОС	Функции элементов ОС	
	Стационарный процесс	Нестационарный процесс
Ядро	Контроль допустимого времени ожидания приема сообщений-откликов от всех ЦВМ, участвующих в обмене данными, выполняемом в синхронном режиме. Запись параметров состояния ВП по завершении обработки КТ, обеспечивающих возможность повторного выполнения прикладных задач при восстановлении ВП с помощью процедуры рестарта	Обнаружение неисправности по признаку превышения допустимого времени ожидания сообщения-отклика от какой-либо ЦВМ, выполняющей межмашинный обмен в синхронном режиме. Управление ресурсами комплекса, корректирующее работу средств межмашинного обмена
Гипервизор	В каждой КТ выполнение шагов обработки, реализующих выбор согласованного значения и контролирующих идентичность (допустимое отклонение) значений результатов решения активных копий прикладных задач в комплексе ЦВМ	Реализация способа восстановления ВП в комплексе (системе) с помощью механизмов, выполняющих каждый шаг обработки КТ на всем интервале восстановления
Средства межмашинного обмена	Установление признаков успешного завершения межмашинного обмена	Установление признаков безуспешного завершения межмашинного обмена с неисправной ЦВМ. Восстановление информации после сбоев шины
Средства встроенного контроля	Подтверждение исправного состояния ЦВМ при выполнении диагностических программ	Обнаружение признаков проявления отказа ЦВМ и выполнение ее самоблокировки

лизу ОС нерезервированных МВС, под гипервизором понималась системная программа, обеспечивающая согласованную работу нескольких ОС в одной или нескольких ЦВМ [21]. В таком понимании гипервизор является элементом ОС, обеспечивающим согласованную работу средств обеспечения отказоустойчивости в каждом комплексе (системе комплексов) в статических (динамических) МВС. Гипервизор является той частью аппаратурно-программных средств обеспечения отказоустойчивости, которая на наиболее высоком уровне выполняет восстановление ВП.

Связи между элементами ОС в МВС с детерминированным и случайным распределением ресурсов организованы существенно различным образом. Поэтому схема связей между элементами ОС рассматривается в последующих разделах при анализе реализации отказоустойчивости в этих системах.

2.3. *Постановка задачи.* Цель работы состоит в определении закономерностей, связывающих способы управления ВП в РВ и его восстановления после проявления неисправностей. Объектом исследования являются основные элементы ОС, прежде всего, ядро и гипервизор. Основу работы составляет гипотеза о закономерном ха-

Таблица 4

Классификация отказоустойчивых управляющих МВС

Существенные параметры	Деление МВС			
	Классы			
Характер распределения ресурсов	Детерминированный		Случайный	
Характер реализации управления прикладными задачами	Чисто системная реализация	Реализация с использованием исполнительной системы параллельных ЯВУ	Чисто системная реализация	Реализация с использованием исполнительной системы параллельных ЯВУ
	Подклассы			
Варианты резервирования и число комплексов, распределение прикладных задач между ними, структура управления межмашинным обменом	1. Статические КПР 2. Статические КРЗ 3. Статические системы из нескольких синхронных КПР 4. Динамические системы из нескольких синхронных КПР	Системы из одного или нескольких синхронных КПР	1. Системы с централизованным управлением межмашинным обменом 2. Системы с распределенным управлением межмашинным обменом	1. Системы с централизованным управлением межмашинным обменом 2. Системы с распределенным управлением межмашинным обменом

рактере связей между ядром и гипервизором в любой отказоустойчивой МВС. *Все рассматриваемые управляющие МВС разбиты по классам и подклассам таким образом, чтобы было очевидно, что для каждого из них существуют принципиально различные способы управления и восстановления ВП* (табл. 4).

Исследование проведено раздельно для МВС с детерминированным и случайным распределением ресурсов. В каждом классе необходимо определить базовый подкласс, в котором реализован наиболее широкий спектр способов восстановления. Далее для каждого класса следует определить общую структуру ОС и последовательно выполнить анализ способов реализации основных элементов во всех подклассах, начиная с базового. Затем проводится сравнительный анализ качества реализации отказоустойчивости МВС с заданным распределением ресурсов между всеми подклассами. Наконец, в заключение необходимо провести сравнительный анализ способов реализации основных элементов ОС, осуществляющих управление ВП в системах с детерминированным и случайным распределением ресурсов, по степени поддержки ими отказоустойчивости МВС.

3. МВС с детерминированным распределением ресурсов

Все отказоустойчивые МВС с детерминированным распределением ресурсов разделены на подклассы по способу управления ВП по следующим признакам: число комплексов в МВС (один или несколько); распределение ЦВМ между комплексами

(статическое или динамическое); вариант резервирования ЦВМ в комплексах (КПР или КРЗ). Практически реализованные МВС образуют 4 подкласса: статический КПР, статический КРЗ, статическая система из нескольких КПР, динамическая система из одного или нескольких КПР и нерезервированных ЦВМ. Базовым подклассом, в котором реализован наиболее широкий спектр способов восстановления, является статический КПР.

3.1. *Структура ОС.* Для всех МВС с детерминированным распределением ресурсов характерно постоянное (резидентное) размещение системных и прикладных программ в памяти каждой ЦВМ, а также явное деление системных программ на группы, реализующие функции каждого элемента ОС. Объясним взаимодействие между элементами ОС с помощью многоуровневой модели [22, 23] для того, чтобы показать характер преобразования информации о неисправности, передаваемой при восстановлении ВП. Пронумеруем уровни в последовательности от высшего (прикладные задачи) к низшему (средства встроенного контроля ЦВМ). Характер выполнения прикладных задач (уровень 1) после окончания восстановления должен оставаться в исправных ЦВМ таким же, каким был до момента возникновения неисправности. Исходные данные для прикладных задач поступают от гипервизора (уровень 2), который "скрывает" от них как ошибочные значения, принимаемые от неисправных ЦВМ, так и уменьшающуюся после их отказов кратность резервирования. В свою очередь, гипервизор выполняет обработку резервированных значений исходных данных и результатов решения активных копий прикладных задач, принимаемых от других ЦВМ с помощью средств межмашинного обмена (уровень 3). Эти средства передают гипервизору либо конкретное значение данных, либо признак безуспешного завершения обмена с неисправной ЦВМ. Согласованную работу средств внутрикомплексного обмена в передающей и принимающей ЦВМ на интервале восстановления обеспечивает ядро (уровень 4), которое также корректирует работу супервизора (уровень 5) с тем, чтобы обеспечить гипервизор достаточной информацией о неисправности. Супервизор осуществляет диспетчеризацию прикладных и системных задач без вмешательства ядра только в стационарном ВП. При восстановлении ВП его работа в исправных ЦВМ комплекса выполняется под управлением ядра. В случае обнаружения неисправности средствами встроенного контроля (уровень 6) до начала восстановления работа супервизора в неисправной ЦВМ должна быть блокирована.

3.2. *Реализация основных элементов ОС в статическом КПР.* В любом статическом КПР все исправные ЦВМ выполняют одинаковую последовательность прикладных задач и обработки КТ. В синхронном КПР все ЦВМ имеют единую службу системного времени, в асинхронном каждая – индивидуальную. Функции всех элементов ОС статического КПР совпадают с приведенными в табл. 2 и 3. Приведем механизмы, используемые для их реализации. При анализе элементов ОС основное внимание уделено тому, КАК механизмы гипервизора выполняют обработку КТ. По механизмам же супервизора и ядра определено только, ЧТО они выполняют и ПОЧЕМУ их работа нужна для поддержки гипервизора. Анализ способов реализации основных элементов ОС содержит следующую последовательность шагов: определение механизмов диспетчеризации прикладных задач и межмашинного обмена; определение механизмов ядра, реализующих синхронный режим межмашинного обмена; анализ способов восстановления на уровне шагов обработки КТ; определение механизмов гипервизора, используемых для реализации каждого шага; анализ характеристик структуры комплекса и ВП, ограничивающих область применения каждого механизма гипервизора; анализ механизмов ядра, реализующих функцию управления ресурсами комплекса в тех способах восстановления, в которых она необходима для поддержки гипервизора; сравнительный анализ качества реализации отказоустойчивости для всех способов восстановления, реализованных в рассматриваемом подклассе МВС.

Естественно, что полностью процедура анализа выполнима только для тех подклассов МВС, в которых могут быть реализованы несколько способов восстановления, причем хотя бы один из них требует управления ресурсами комплекса. Этими свойствами обладает только статический КПР. Во всех остальных подклассах, рассматриваемых в разделах, начиная с 3.3, отказоустойчивость реализована, как правило, единственным способом, без управления ресурсами комплекса (не выполняются шаги 4 и 5). Также не всегда проявляется специфика реализации супервизора (шаг 1). Таким образом, общими для анализа реализации основных элементов ОС во всех управляющих отказоустойчивых МВС являются только шаги 2 и 3 процедуры.

3.2.1. *Диспетчеризация прикладных задач и обмена данными.* В любой МВС РВ супервизор осуществляет диспетчеризацию прикладных задач и обмена данными либо по времени, либо по событиям, либо смешанным способом. Во всех МВС с детерминированным распределением ресурсов используется диспетчеризация либо по времени, либо смешанным способом. Определим свойства управления ВП, обусловившие необходимость выполнять в любой управляющей МВС диспетчеризацию хотя бы части прикладных задач по времени. Затем дадим определения механизмов диспетчеризации. МВС с детерминированным распределением ресурсов, как правило, применяются в системах управления техническими объектами. *В таких системах большинство прикладных задач имеет циклический характер выполнения с периодами, кратными некоторой минимальной величине, именуемой малым циклом (МЦ)* [24]. В МЦ выполняются одна или несколько задач. Во всех супервизорах, реализующих смешанный способ диспетчеризации, период генерации сигналов таймера равен МЦ. Сигналы таймера используются для активизации первой задачи в группе выполняемых в МЦ. В супервизорах, реализующих способ диспетчеризации по времени, период генерации сигналов таймера равен величине кванта времени, предоставляемого для выполнения каждой задачи.

Для диспетчеризации прикладных задач в МВС с детерминированным распределением ресурсов используется либо один из трех базовых механизмов (временной, списковый, внешних прерываний), либо для разных задач используются различные механизмы. Временной механизм реализует способ диспетчеризации по времени, остальные – смешанный способ. Дадим определения базовых механизмов. *Во временном механизме выполнение каждой задачи инициируется сигналом таймера. В каждом интервале выполняется только одна, заранее определенная задача, поэтому величина интервала должна быть не меньше, чем значение оценки максимальной длительности выполнения (по множеству задач, выполняемых в ЦВМ). В списковом механизме для управления ВП используются списки задач. Каждый список задает последовательность выполнения задач в МЦ без указания времени, выделяемого каждой задаче. В механизме внешних прерываний выполнение каждой прикладной задачи в МЦ, кроме первой, инициируется соответствующим сигналом. Последовательность выполнения задач заранее неизвестна. Все задачи должны быть упорядочены по приоритетам. Если инициируется высокоприоритетная задача, она прерывает выполнение низкоприоритетной, перемещая ее в очередь.*

3.2.2. *Механизмы ядра, реализующие синхронный режим внутрикомплексного обмена.* В любой МВС с детерминированным распределением ресурсов в памяти каждой ЦВМ хранится собственная копия средств реализации синхронного режима, используемая относительно независимо от остальных ЦВМ. В известных статических КПР используются два способа реализации синхронного режима – механизмы синхронизации по времени и по событиям. В механизме синхронизации по времени периодически (по сигналам от таймера) все исправные ЦВМ переводятся в одно и то же состояние ВП, исходное для выполнения прикладной задачи, обмена данными или обработки КТ [25]. В каждой ЦВМ может быть использован ее собственный таймер, однако в комплексе должна быть организована единая служба системного

времени. Понятно, что диспетчеризация прикладных задач в таком комплексе должна быть реализована с помощью временного механизма. Механизм синхронизации по событиям ориентирован на другие базовые механизмы диспетчеризации – списковый и внешних прерываний. Его функция состоит в одинаковом для всего комплекса упорядочении обработки внешних событий, независимо (асинхронно) возникающих в каждой ЦВМ. В разные ЦВМ комплекса в различной последовательности поступают из внешней среды сигналы одних и тех же внешних прерываний (например, вследствие асинхронной работы резервированных устройств, являющихся источниками этих сигналов), которые в одинаковой последовательности должны быть ими обработаны. В [26] приведен алгоритм, реализующий механизм синхронизации по событиям.

3.2.3. *Реализация способов восстановления. Упорядоченный набор значений результатов решения активных копий прикладной задачи, формируемый в КТ одинаковым образом всеми ЦВМ КПР, назовем исходным набором (ИН). Место расположения каждого элемента ИН интерпретируется при обработке КТ как окно приема значения результата решения активной копии задачи от конкретной ЦВМ по определенной шине. Отсутствие принятого значения в течение заданного времени ожидания, появление рассогласованного значения или признака безуспешного завершения обмена интерпретируется, как проявление неисправности ЦВМ или шины.*

По признаку идентичности или неидентичности значений результатов решения задачи, передаваемых от неисправной ЦВМ во все исправные, различают "дружественную" (*nonbyzantine*) и "враждебную" (*byzantine*) формы проявления неисправности [27]. *ЦВМ, находящаяся в состоянии неисправности, проявляющейся в "дружественной" форме, передает всем исправным одно и то же рассогласованное значение собственного результата решения задачи. В состоянии же неисправности, проявляющейся во "враждебной" форме, ЦВМ может передать разным исправным различные значения собственного результата вычислений, в том числе и совпадающие с согласованным.*

Процедура обработки КТ в КПР состоит из следующих шагов [11]: преобразование ИН, при котором часть "враждебных" форм проявления неисправностей переходят в "дружественную", остальные – в скрытую форму; выбор согласованного значения в ИН; обнаружение неисправности, проявляющейся в виде рассогласованных значений одного или нескольких элементов ИН; локализация неисправной ЦВМ или шины; интерпретация характера проявления неисправности, как сбоя или отказа; восстановление ВП после сбоя; блокирование отказавшей ЦВМ (шины) и восстановление ВП после отказа.

При анализе способов восстановления ВП, реализованных в КПР, *под локализацией понимается шаг обработки КТ, в котором по месту расположения рассогласованных элементов ИН определяется порядковый номер резервированной ЦВМ (шины), неисправность которой вызвала их возникновение. Значение этого номера будем называть результатом локализации.* При "дружественной" форме проявления неисправности места расположения рассогласованных элементов в ИН каждой исправной ЦВМ совпадают, при "враждебной" – различаются. Для того чтобы все исправные ЦВМ КПР сформировали идентичные результаты локализации "враждебной" неисправности в момент ее проявления, требуется выполнить первый шаг обработки – преобразование ИН. При реализации способов восстановления ВП, используемых либо только при "дружественных" формах проявления неисправностей, либо при "враждебных" с локализацией неисправной ЦВМ в течение нескольких КТ с момента обнаружения, обработка КТ начинается со второго шага.

Любой способ восстановления реализуем только при одиночном проявлении неисправностей. *Под одиночным понимается проявление неисправностей, при котором в интервале выполнения локализации одного неисправного устройства вероятность проявления неисправности в любом другом пренебрежимо мала.*

Будем различать группы способов восстановления ВП в КПР по следующим признакам.

1. Форма проявления неисправности ("дружественная" или "враждебная").

2. Уровень резервирования аппаратуры, рассматриваемый как неделимое целое (*atomic*) при интерпретации результата локализации, – каналы или устройства. *Каналом называем жесткую связку ЦВМ+шина, в которой гипервизор не может раздельно интерпретировать признаки проявления неисправностей ЦВМ и шины.* Резервирование на уровне каналов реализуется путем постоянного распределения полномочий управления шинами внутрикомплексного обмена между ЦВМ. *Устройствами называем ЦВМ и шину в отдельности при анализе тех способов восстановления, в которых признаки проявления их неисправностей различимы.* Резервирование на уровне устройств реализуется за счет циклической передачи полномочий управления любой шиной между всеми ЦВМ комплекса, интерпретируемыми гипервизором как исправные.

3. Объем всех "враждебных" форм проявления неисправностей, которые могут быть локализованы при использовании способа восстановления.

Результат локализации "дружественных" неисправностей на уровне устройств, так же как и всех форм проявления "враждебных" на уровне каналов, может быть сформирован при использовании существенно различных способов управления ресурсами комплекса. Поэтому при "дружественных" формах проявления неисправностей результат локализации может быть сформирован на уровне либо каналов, либо устройств, тогда как при "враждебных" – только на уровне каналов.

Для реализации способов восстановления ВП, в которых информация о признаках проявления неисправности накапливается в течение нескольких КТ с момента ее проявления, используются следующие режимы управления ресурсами комплекса: переход вперед (ПВ) и возврат назад (ВН) [11]. *Режим управления ресурсами комплекса, при котором сохраняется поступательный характер развития ВП на интервале его восстановления, именуется ПВ (forward error recovery, roll ahead). Режим повторного выполнения ВП при его восстановлении принято называть ВН или откатом (backward error recovery, roll back).*

Группы способов восстановления ВП в КПР различаются наборами исходных данных, необходимых для обработки КТ (табл. 5).

Полный перечень наборов данных для обработки КТ включает в себя: ИН результатов решения активной копии прикладной задачи в каждой ЦВМ; вспомогательный набор признаков успешного/безуспешного завершения внутрикомплексного обмена по каждой шине, установленных по завершении цикла передачи полномочий управления шиной между всеми ЦВМ; вспомогательные наборы признаков обнаружения рассогласованных элементов в ИН и неоднозначных результатов локализации "враждебной" неисправности, сформированных в каждой ЦВМ комплекса независимо от остальных. При реализации любого способа восстановления используется только часть наборов. Пояснения по каждому набору приведем при рассмотрении механизмов гипервизора.

Анализ каждого способа восстановления проведем по шагам обработки КТ. Для каждого шага приведем все реализующие его механизмы гипервизора. Возможность реализации любого механизма зависит от следующих характеристик структуры КПР и управления ВП: минимально необходимого числа активных копий задачи (совпадающей с кратностью резервирования ЦВМ); режима внутрикомплексного обмена; режима управления ресурсами комплекса.

Будем использовать буквенно-индексные обозначения механизмов гипервизора. Буквенная часть указывает на шаг процедуры, индексная – на порядковый номер в множестве механизмов, которые могут выполнять этот шаг. Обозначим шаги процедуры: α – преобразование ИН, при котором часть "враждебных" форм проявления неисправностей переходят в "дружественную", остальные – в скрытую форму,

Таблица 5

Способы восстановления ВП в КПР и исходные данные для обработки КТ

Исходные данные для обработки КТ	Способы восстановления ВП			
	С локализацией только "дружественных" форм проявления неисправностей на уровне		С локализацией всех "дружественных" и части "враждебных" форм проявления неисправностей на уровне каналов	С локализацией любых форм проявления "дружественных" и "враждебных" неисправностей на уровне каналов
	каналов	устройств		
Исходный набор результатов решения активных копий прикладной задачи (вектор A_n)	+	+	−	−
Исходный набор результатов решения активных копий прикладной задачи (матрица B_n)	−	−	+	+
Вспомогательный набор признаков успешного/безуспешного завершения внутрикомплексного обмена (матрица C_n)	−	+	−	−
Вспомогательные наборы: признаков обнаружения рассогласованных элементов во всех B_n (вектор D_n); неоднозначных результатов локализации "враждебной" неисправности (векторы E_n и F_n)	−	−	−	+

β – выбор согласованного значения в ИН, γ – обнаружение неисправности, δ – ее локализация, ε – интерпретация характера проявления (сбой или отказ), ζ – восстановление после сбоя, η – блокирование отказавшей ЦВМ и восстановление после отказа.

3.2.3.1. *Восстановление с локализацией "дружественных" неисправностей на уровне каналов.* Эти способы обладают следующими свойствами: каждый шаг обработки КТ может быть выполнен разными механизмами гипервизора; в зависимости от используемого механизма локализации неисправности восстановление реали-

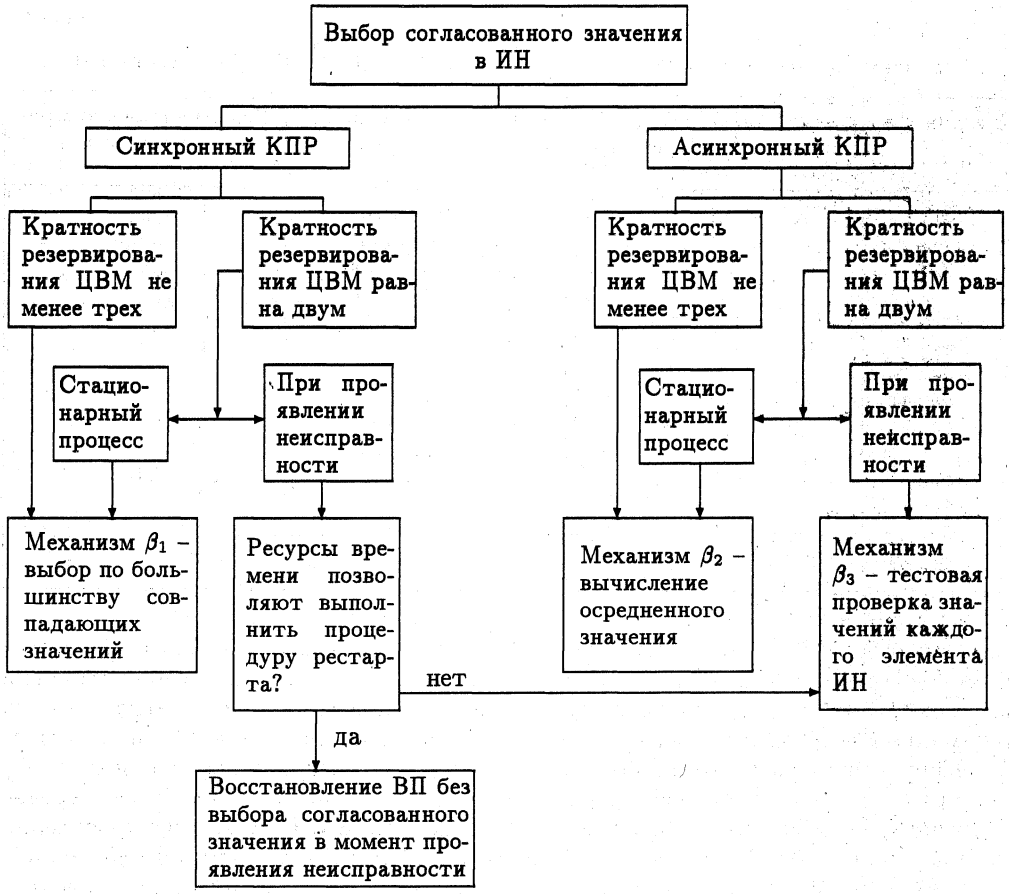

Рис. 2. Механизмы выбора согласованного значения в ИН при обработке КТ в КПР

зуемо с помощью одно- или многократной последовательной обработки КТ. Область применения каждого механизма ограничена определенными характеристиками структуры КПР и управления ВП. По каждому шагу обработки КТ приведем все возможные варианты его реализации (механизмы гипервизора) с указанием характеристик структуры КПР и управления ВП, ограничивающих область их применения.

Выбор согласованного значения в ИН. Структура ИН, формируемого в n-й ЦВМ, имеет вид вектора A_n размерности N ($n = \overline{1,N}$, где N – кратность резервирования ЦВМ). Любой i-й элемент вектора A_n ($i = \overline{1,N}$) содержит либо значение результата решения активной копии задачи в i-й ЦВМ, принятый в n-ю (при $i \neq n$), либо признак безуспешного завершения обмена между i-й и n-й ЦВМ. Во всех исправных ЦВМ значения i-го элемента A_n одинаковые. Способ выбора согласованного значения зависит от кратности резервирования ЦВМ и от режима внутрикомплексного обмена (рис. 2, табл. 6).

В синхронном комплексе значения элементов A_n, принятые от исправных ЦВМ, должны совпадать, в асинхронном – могут различаться не более, чем на заданную величину. В том случае, если кратность резервирования не менее трех, большинство элементов ИН соответствуют результатам вычислений в исправных ЦВМ. При этом для выбора согласованного значения в синхронном КПР используется механизм β_1 – по большинству совпадающих значений, в асинхронном – β_2 – путем вычисления осредненного значения. Осредненное значение может быть вычислено различ-

Таблица 6

Зависимость механизмов от характеристик структуры КПР и управления ВП

Механизмы	Минимально необходимая кратность резервирования ЦВМ	Режим внутрикомплексного обмена	Режим управления ресурсами комплекса
β_1	3	синхронный	–
β_2	3	асинхронный	–
β_3	2	синхронный или асинхронный	–
γ_1	3	синхронный	–
γ_2	2	–"–	–
γ_3	3	асинхронный	–
γ_4	2	–"–	–
δ_1	2	синхронный или асинхронный	–
δ_2	2	синхронный	ВН
ζ_1	2	–"–	–
η_1	3	–"–	–
η_2	–	–	–

ными способами [28]: вычислением среднего арифметического по всем элементам A_n; по центральному элементу в векторе A'_n, являющемся результатом сортировки A_n по возрастанию или убыванию значений его элементов (при нечетном N); по минимальному (максимальному) из двух центральных элементов A'_n (при четном N); вычислением среднего арифметического по двум центральным элементам A'_n (при четном N). В том случае, если кратность резервирования равна двум, перечисленные механизмы применимы только в стационарном ВП. В момент проявления неисправности выбор согласованного значения может быть выполнен только одним способом – с помощью механизма β_3 – тестовых проверок значений каждого элемента. Этот механизм является единственно возможным в дублированном асинхронном КПР [29 – 31]. В синхронном же КПР кроме него допустимо восстановление с помощью процедуры рестарта без выбора согласованного значения в момент проявления неисправности. Согласно экспертным оценкам [32], вероятность выбора "правильного" результата решения задачи с помощью $\beta_1(\beta_2)$ существенно выше, чем с помощью β_3.

Обнаружение неисправности по значениям элементов ИН. Выполнимо с помощью четырех механизмов. Одиночная неисправность в синхронном КПР проявляется в виде несовпадения значения одного элемента A_n с остальными, в асинхронном – как недопустимая величина отклонения одного элемента от остальных. Если кратность резервирования не менее трех, то в синхронном (асинхронном) КПР для обнаружения неисправности используется $\gamma_1(\gamma_3)$ – контроль на совпадение (на допустимое отклонение) каждого элемента ИН с согласованным значением. Если же кратность резервирования ЦВМ равна двум, то контроль на совпадение (на допустимое отклонение) осуществляется между самими элементами ИН с помощью $\gamma_2(\gamma_4)$. Вероятность успешного обнаружения неисправности любым механизмом одинакова (рис. 3, табл. 6).

Локализация неисправного канала. Локализация "дружественных" неисправностей на уровне каналов может быть выполнена одним из двух механизмов, применяемым в зависимости от того, реализован ли выбор согласованного значения в момент проявления неисправности (табл. 6). Если согласованное значение было выбрано любым из $\beta_1 - \beta_3$, то локализация осуществляется с помощью δ_1 – по

Рис. 3. Механизмы обнаружения "дружественных" неисправностей по значениям элементов ИН в КПР

месту расположения единственного рассогласованного элемента в A_n. Если же для восстановления применяется процедура рестарта, то локализация выполняется с помощью δ_2 – контроля на совпадение с эталонным значением результата решения прикладной задачи, записанным в память в КТ, предшествующей моменту проявления неисправности. Поясним способ реализации процедуры рестарта и работу δ_2.

В стационарном ВП после завершения обработки каждой КТ в память ЦВМ записывается информация двух видов [33]: успешно прошедшие контроль на совпадение согласованные значения элементов ИН, интерпретируемые при выполнении восстановления как эталонные; значения параметров состояния ВП, обеспечивающие возможность рестарта от данной КТ.

Тип КТ, в котором осуществляется запись информации как первого, так и второго вида, назовем точками хранения. Именно точка хранения является моментом начала выполнения процедуры рестарта. *Тип КТ, в котором осуществляется запись информации только первого вида, назовем точками сравнения.* В зависимости от условия завершенности восстановления точки сравнения могут быть либо промежуточными, либо конечными на интервале восстановления. Естественно, что точки сравнения выполняются чаще, чем точки хранения. Существенно, что любая операция обмена с абонентами должна содержать точку сравнения. Это требование обеспечивает возможность временной блокировки обмена с абонентами при повторном выполнении прикладных задач во время восстановления. Пояснения по реализации временной блокировки обмена приведены в разделе 3.2.4.

После того, как при обработке КТ обнаружено проявление неисправности, выполняется процедура рестарта. При этом в памяти каждой ЦВМ восстанавливаются значения параметров состояния ВП, записанные в последней точке хранения до проявления неисправности, после чего осуществляется повторное выполнение прикладных задач. Интервал между точкой хранения, являющейся начальной в процедуре рестарта, и моментом проявления неисправности, должен содержать не менее одной точки сравнения, чтобы при повторном выполнении задач хотя бы однократно выполнить контроль на совпадение с эталонными значениями, записанными до

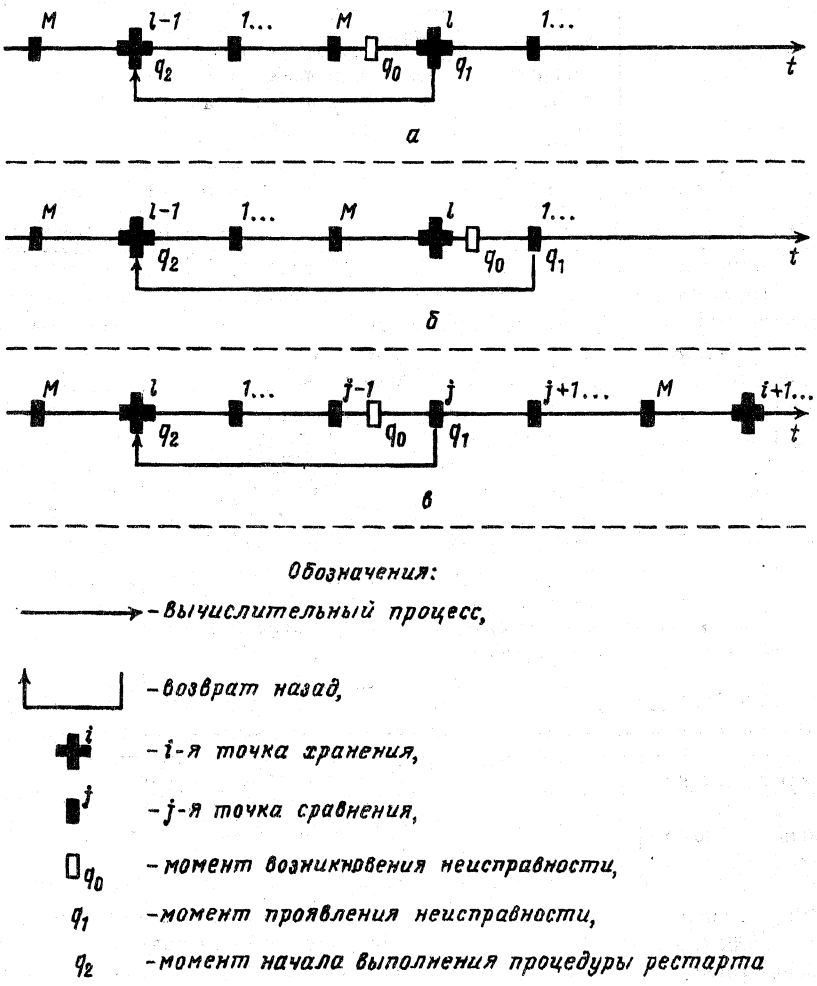

Рис. 4. Зависимость начала процедуры рестарта от типа КТ, в которой проявляется неисправность. *а* – момент проявления неисправности совпадает с точкой хранения; *б* – момент проявления неисправности совпадает с первой точкой сравнения; *в* – момент проявления неисправности совпадает с j-й ($j \geqslant 2$) точкой сравнения

возникновения неисправности. На рис. 4 приведены все возможные варианты расположения во времени моментов возникновения неисправности, ее проявления и начала выполнения процедуры рестарта на примере ВП, в котором интервал между любыми соседними точками хранения содержит одинаковое число точек сравнения $M \geqslant 2$.

В каждой КТ на интервале восстановления каждая ЦВМ осуществляет контроль совпадения собственного результата вычислений с эталонным значением. ЦВМ, обнаружившая несовпадение собственного результата выполнения прикладной задачи с эталонным значением, интерпретируется как неисправная. Если же при повторном выполнении всех точек сравнения во время восстановления обеими ЦВМ не было обнаружено несовпадений с эталонными значениями, то этот результат интерпретируется как успешное восстановление после сбоя и комплекс продолжает работу без деградации.

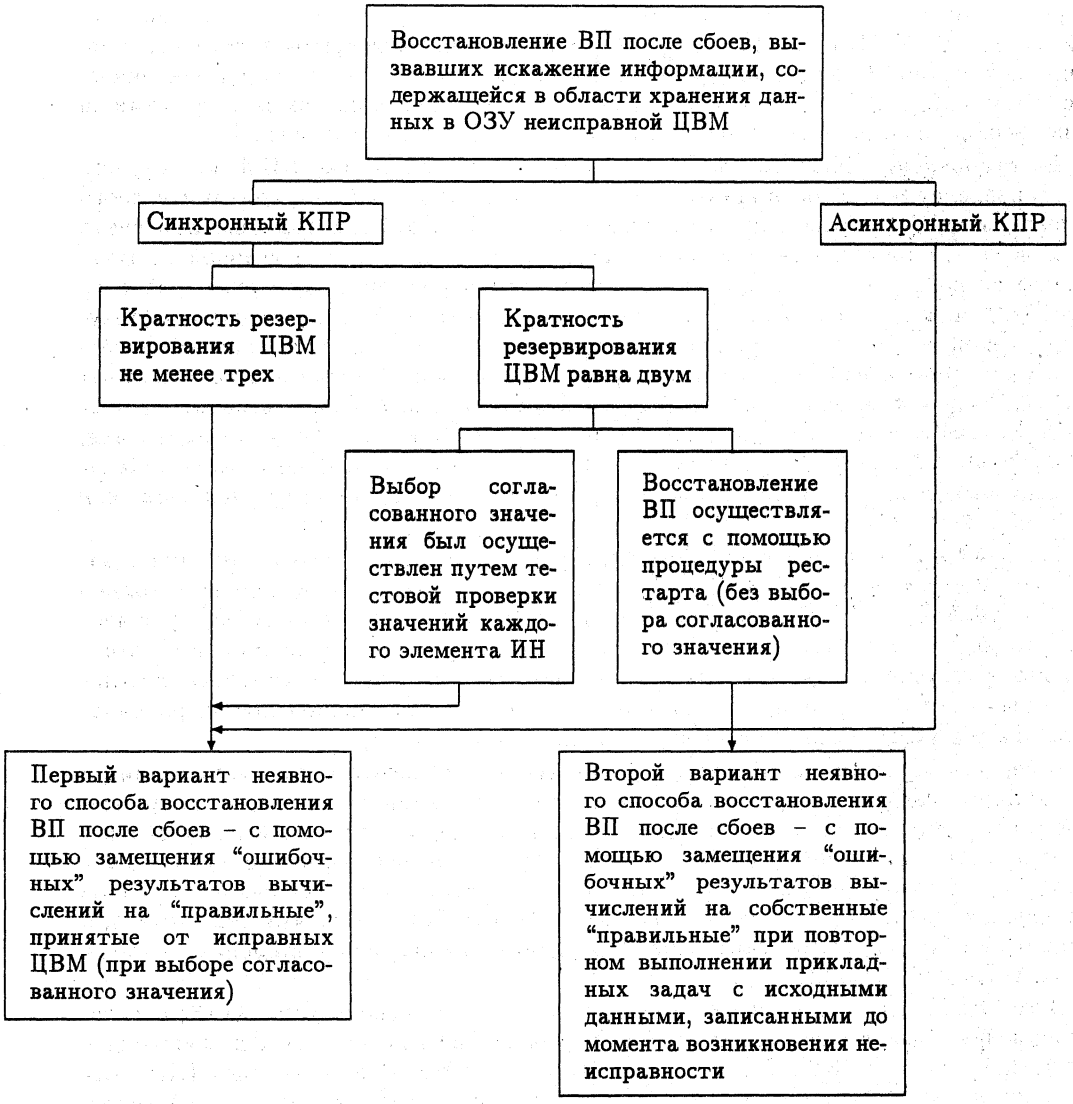

Рис. 5. Условия применения неявного способа восстановления ВП после сбоев ЦВМ в КПР

Вероятность успешной локализации неисправности с помощью δ_1 в КПР с кратностью резервирования ЦВМ не менее трех выше, чем любым механизмом, используемым в дублированном комплексе.

Интерпретация характера проявления неисправности. Этот шаг обработки КТ выполняется единственным механизмом ε_1, причем возможность его использования не зависит от управления ВП в КПР. Интерпретация характера проявления неисправности как сбоя или отказа осуществляется по значению счетчика выполненных подряд и завершившихся безуспешно попыток восстановления после сбоя, локализованного в одном и том же канале. В каждой КТ производится не более одной попытки восстановления, причем даже в случае безуспешного ее завершения продолжается выполнение ВП (за счет успешно выбранного согласованного значения). До тех пор, пока счетчик не превысит порогового значения, установленного для изменения интерпретации характера проявления неисправности, она интерпретируется как сбой,

в противном случае – как отказ. Счетчик обнуляется при подтверждении исправной работы ЦВМ. Предельно допустимое значение числа попыток восстановления, начиная с которого неисправность интерпретируется как отказ (пороговое значение счетчика), устанавливается в зависимости от соотношения между затратами на выполнение одной попытки и ресурсом, выделенным на восстановление.

Восстановление после сбоев ЦВМ. Рассмотрим сбои только ЦВМ потому, что восстановление после сбоев шины осуществляется средствами внутрикомплексного обмена, скрытым образом от остальной части ОС. Сбой ЦВМ в конечном счете проявляется в виде искажения информации, хранимой в ее оперативной памяти, причем способы восстановления зависят от типа информации, которую требуется восстанавливать, – постоянная (программы и константы) или изменяемая (текущая) в процессе решения прикладных задач. *Способ восстановления после сбоев, реализуемый с помощью данных внутрикомплексного обмена или собственных результатов вычислений, используемых в механизмах выбора согласованного значения и локализации неисправности, назовем неявным. Способ восстановления, требующий специальной передачи информации от исправных ЦВМ в восстанавливаемую только после интерпретации характера проявления неисправности, как сбоя, назовем явным.*

Если в ЦВМ для хранения программ и констант используется постоянное запоминающее устройство (ПЗУ), то после сбоев требуется восстанавливать только область хранения данных в оперативном запоминающем устройстве (ОЗУ), для чего достаточно использовать неявный способ восстановления. В зависимости от того, используются данные внутрикомплексного обмена или только собственные результаты вычислений по данным, записанным до проявления неисправности, различаются два варианта неявного способа восстановления (рис. 5).

В первом варианте замещение в ОЗУ "ошибочных" результатов вычислений на "правильные" выполняется с помощью данных внутрикомплексного обмена, при выборе согласованных значений. Для его реализации требуются условия, которые необходимы для механизмов β_1, β_2 и β_3 (табл. 6). Во втором варианте восстановление информации осуществляется с использованием только собственных результатов выполнения прикладных задач с заведомо корректными исходными данными при выполнении процедуры рестарта. Для его реализации требуются условия, которые необходимы для механизма δ_2.

В тех ЦВМ, где для хранения программ и констант используется ОЗУ, восстановление после сбоя может быть выполнено только явным способом: с помощью ζ_1 – копирования памяти (КП). Этот механизм применим только в синхронном КПР, так как в результате его выполнения исправные и восстанавливаемая ЦВМ переводятся в одно и то же состояние ВП, от которого они одновременно должны возобновить выполнение прикладных задач (рис. 6).

В асинхронном комплексе механизмы восстановления информации в области хранения программ и констант после сбоев ЦВМ отсутствуют. Благодаря успешному выбору согласованных значений исправными ЦВМ в случае существенных затрат времени ζ_1 может быть использован не сразу, в момент проявления неисправности, а позже, в ближайший момент минимальной вычислительной нагрузки комплекса. Таким образом, единственным явно используемым средством восстановления после сбоев ЦВМ является ζ_1. Для того чтобы выполнить ζ_1, требуется передать группу сообщений в восстанавливаемую ЦВМ. Достаточность ресурсов времени для реализации ζ_1 зависит от следующих параметров: пропускной способности шины и ее надежности, а также от объема восстанавливаемых данных и допустимых сроков их передачи [20, 34].

Блокирование отказавшего канала. Реализуемо двумя способами: с помощью аппаратурных средств, работающих по сигналам, принятым от исправных ЦВМ (η_1) или с помощью программных средств неисправной, выполняющих ее самобло-

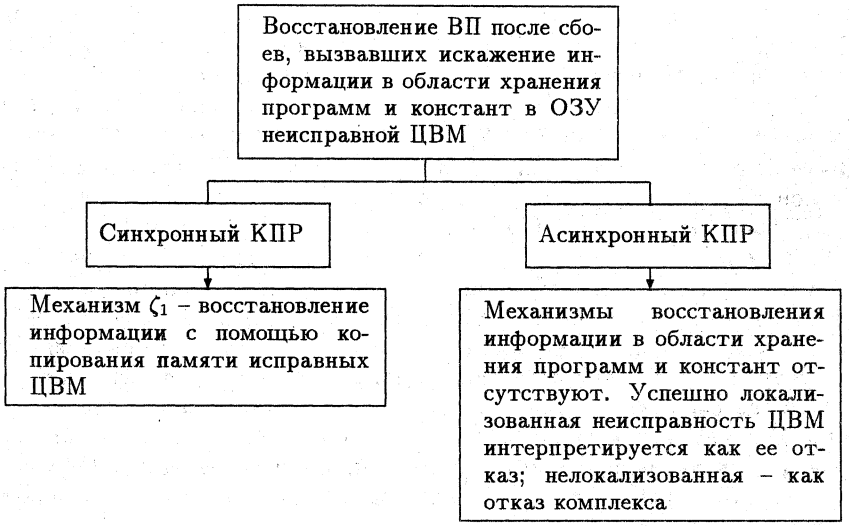

Рис. 6. Условия применения явного способа восстановления ВП после сбоев ЦВМ в КПР

кировку (η_2). Механизм η_1 применим только в синхронном КПР при кратности резервирования ЦВМ не менее трех. Механизм η_2 работает автономно (независимо от остальных ЦВМ) и потому не зависит от управления ВП. Механизм η_1 является более надежным, чем η_2, поскольку он предотвращает ложное блокирование исправной ЦВМ или задержку в блокировании отказавшей. Как правило, для повышения надежности блокирования используются средства аппаратурной поддержки, срабатывающие при синхронном получении большинства сигналов, переданных исправными ЦВМ [32, 35].

3.2.3.2. *Восстановление с локализацией "дружественных" неисправностей на уровне устройств.* Для успешной локализации "дружественной" неисправности способ восстановления требует последовательной обработки нескольких КТ с момента ее проявления. Ядро поддерживает работу гипервизора с помощью управления ресурсами комплекса (режим ПВ) [36 – 40]. Для этого требуется, чтобы КПР был синхронным с кратностью резервирования ЦВМ не менее трех. Синхронный режим необходим для того, чтобы реализовать механизм циклического перераспределения полномочий управления шинами между ЦВМ в каждой КТ, позволяющий сформировать вспомогательный набор C_n признаков успешного/безуспешного завершения внутрикомплексного обмена (табл. 5). C_n имеет структуру прямоугольной матрицы размерности $M \times N$, где $N(M)$ – кратность резервирования ЦВМ (шин). c^n_{mj} – элемент матрицы C_n ($j = \overline{1,N}$; $m = \overline{1,M}$) – содержит значение признака успешного/безуспешного приема данных из j-й в n-ю ЦВМ по m-й шине. Кратность резервирования, не менее трех, необходима для успешного выбора согласованного значения в ИН на всем интервале восстановления, а также для реализации процедуры мажоритарного выбора в δ_3, рассмотренном ниже. C_n формируется в нескольких КТ с момента проявления неисправности. В разных КТ каждая ЦВМ должна использовать разные шины для передачи данных во внутрикомплексном обмене, чтобы обеспечить гипервизор дополнительной информацией о признаках проявления неисправности, необходимой для ее локализации. Формирование C_n завершается после окончания цикла передачи полномочий управления шинами между ЦВМ комплекса, интерпретируемыми гипервизором как исправные. Для реализации рассматриваемого способа восстановления необходимо по каждому допустимому соотношению

числа исправных ЦВМ и шин составить расписание циклической передачи полномочий, чтобы устранить возможность конфликта, а также обеспечить одновременность начала и окончания формирования C_n во всех ЦВМ. Передача полномочий управления шинами может осуществляться либо постоянно в каждой КТ (*SIFT, SAFE* [20, 39]), либо быть инициированным в момент проявления неисправности (*FTMP* [36]). Начиная с первой и до предпоследней КТ на интервале формирования C_n процедура ее обработки состоит из трех шагов: выбора согласованного значения в ИН (β_1), обнаружения неисправности (γ_1) и записи в C_n значений признаков завершения внутрикомплексного обмена. В последней КТ процедура ее обработки состоит из шести шагов: выбора согласованного значения (β_1), обнаружения неисправности (γ_1), локализации неисправной ЦВМ или шины (δ_3), интерпретации характера проявления неисправности (ε_1), восстановления после сбоя (неявный способ или ζ_1, раздел 3.2.3.1), блокирования отказавшего устройства (если неисправна ЦВМ – η_1, если неисправна шина – исключение ее из используемых в обмене). Таким образом, все шаги обработки КТ, кроме локализации, реализованы теми же механизмами, которые использованы в способе восстановления с локализацией "дружественных" неисправностей на уровне каналов (в той части спектра, которые применимы в синхронном КПР с кратностью резервирования ЦВМ не менее трех). Рассмотрим механизм δ_3.

Локализация неисправной ЦВМ (шины). Выполнение δ_3 состоит из двух шагов. На первом шаге C_n преобразуется в два вектора – признаков проявления неисправностей ЦВМ (V_n) и шин (W_n), размерностью соответственно N и M. На втором шаге определяется, в каком из них содержится рассогласованный элемент и его место расположения. Преобразование C_n реализуется с помощью мажоритарного выбора по большинству совпадающих значений признаков успешного/безуспешного завершения внутрикомплексного обмена, содержащихся в каждом столбце (результатом является V_n) и в каждой строке (результатом является W_n). Естественно, что из мажоритарного выбора должны быть исключены элементы C_n, соответствующие ранее блокированным (отказавшим) ЦВМ и шинам. Пусть номер неисправного устройства равен f. Тогда, если неисправна f-я ЦВМ, то все элементы только f-го столбца C_n содержат значения признака безуспешного завершения обмена. Аналогично, если неисправна f-я шина, то такие же значения содержатся только во всех элементах f-й строки C_n. Все остальные элементы C_n содержат значения признака успешного завершения обмена. В результате преобразования C_n только в одном векторе (либо в V_n, либо в W_n) и только в одном его элементе содержится значение признака безуспешного завершения обмена. Номер этого элемента совпадает с номером неисправной ЦВМ (шины), а сам вектор, содержащий его, указывает, какое именно устройство (ЦВМ или шина) неисправно. Условия, от которых зависит применимость δ_3: минимально необходимая кратность резервирования ЦВМ равна трем; режим внутрикомплексного обмена – синхронный; режим управления ресурсами комплекса – ПВ.

3.2.3.3. *Восстановление с локализацией любых форм проявления "дружественных" и части форм проявления "враждебных" неисправностей на уровне каналов.* Рассматриваемый способ является комбинированным, состоящим из двух частей – преобразования "враждебных" форм проявления неисправностей в "дружественную" или скрытую форму с помощью α_1, реализующего алгоритм взаимного информационного согласования (ВИС) (*Byzantine Agreement*) [27, 41], и восстановления с локализацией "дружественных" неисправностей на уровне каналов (раздел 3.2.3.1). Способ реализуется при обработке одной КТ с момента проявления неисправности. Примером является система *FTP*, состоящая из одного синхронного КПР с кратностью резервирования ЦВМ, равной четырем, в которой механизм α_1 реализован аппаратурно [25, 42, 43]. Процедура обработки КТ выполняется следующими механизмами: преобразование ИН из матрицы B_n в вектор A''_n, при котором

некоторые (наиболее вероятные) "враждебные" формы проявления неисправностей становятся "дружественными", остальные – скрытыми (α_1); выбор согласованного значения в A''_n (β_1); обнаружение неисправности (γ_1); локализация неисправного канала (δ_1); интерпретация проявления неисправности (ε_1); восстановление после сбоя (неявный способ или ζ_1, раздел 3.2.3.1); блокирование отказавшего канала (η_1). Таким образом, все шаги обработки КТ, за исключением первого, реализуются теми же механизмами, что и в синхронном КПР с кратностью резервирования ЦВМ не менее четырех при восстановлении с локализацией "дружественных" неисправностей на уровне каналов.

Преобразование ИН, при котором некоторые "враждебные" формы проявления неисправностей становятся "дружественными", остальные – скрытыми. Работу α_1 можно разделить на два этапа: формирование ИН в виде матрицы B_n; преобразование B_n в вектор A''_n, имеющий структуру, аналогичную A_n.

Матрица B_n имеет размерность $N \times N$ и формируется в каждой (n-й) ЦВМ в двух раундах внутрикомплексного обмена. В первом раунде каждая ЦВМ передает во все остальные результаты собственных вычислений, во втором из любой i-й ЦВМ в n-ю ($i, n = \overline{1,N}, n \neq i$) передаются результаты вычислений, полученные i-й в первом раунде от всех остальных ЦВМ, кроме передающей и принимающей. Матрица B_n имеет следующую структуру: на главной диагонали расположены элементы b^n_{ii} , являющиеся результатами первого раунда; в n-й строке и n-м столбце находится один элемент b^n_{nn}, соответствующий собственному результату вычислений; в остальных i-х строках (j-х столбцах) содержатся элементы b^n_{ij} ($i, j = \overline{1,N}$; $i \neq n$; $j \neq n$). В i-й строке вне главной диагонали содержатся результаты вычислений i-й ЦВМ, принятые от всех j-х. В j-м столбце содержатся результаты вычислений всех i-х ($j \neq n$, $j \neq i$) ЦВМ, принятые от j-й во втором раунде.

Пусть f – номер неисправной ЦВМ. Во всех строках B_n, кроме f-й, содержится не более одного рассогласованного элемента (он может появиться во втором раунде при передаче результата i-й через f-ю ЦВМ). В f-й строке B_n для любых $n \neq f$ содержится одно и то же количество рассогласованных элементов, равное числу исправных ЦВМ, которые в первом раунде приняли от неисправной значение ее результата, интерпретируемое как "ошибочное". Преобразование B_n в A''_n состоит в мажоритарном выборе в каждой строке B_n по большинству совпадающих значений. Во всех элементах A''_n, кроме f-го, должны содержаться только "правильные" (совпадающие с согласованным) значения. Значение f-го элемента A''_n может быть либо "правильным", либо "ошибочным". Если в первом раунде не менее чем $N/2$ исправных ЦВМ приняли от неисправной "ошибочное" значение, оно перейдет в A''_n, т.е. "враждебная" форма проявления неисправности преобразуется в "дружественную". Если же в первом раунде неисправная ЦВМ передала "ошибочное" значение менее чем $N/2$ исправным ЦВМ, то f-й элемент A''_n будет иметь "правильное" значение, т.е. неисправность перейдет в скрытую форму. Условия, необходимые для реализации α_1: минимально необходимая кратность резервирования ЦВМ равна четырем; режим внутрикомплексного обмена – синхронный.

3.2.3.4. *Восстановление с локализацией любых форм проявления "дружественных" и "враждебных" неисправностей на уровне каналов.* Основу восстановления составляет предложенный в [44] механизм локализации, требующий последовательной обработки нескольких КТ с момента проявления неисправности. Так же, как и при восстановлении с локализацией "дружественных" неисправностей на уровне устройств (раздел 3.2.3.2), для поддержки гипервизора требуется управление ресурсами комплекса в режиме ПВ. Комплекс должен быть синхронным с кратностью резервирования ЦВМ не менее четырех, чтобы можно было реализовать алгоритм ВИС. Этот алгоритм используется при обработке КТ как средство обеспечения приема одинаковых значений от неисправной ЦВМ во все исправные, при формировании вспомогательных наборов D_n, E_n и F_n (табл. 5). Кроме того, синхронный режим

внутрикомплексного обмена необходим для управления ресурсами комплекса. Число процедур обработки КТ, в течение которых всеми исправными ЦВМ должен быть сформирован результат локализации, различается для разных форм проявления неисправности. При любой форме проявления гарантирована одновременность начала и окончания локализации неисправности всеми исправными ЦВМ. До тех пор, пока локализация не завершена, в каждой КТ выполняются следующие шаги обработки: выбор согласованного значения в ИН (механизм, сходный с β_1); обнаружение неисправности (γ_5); фрагмент локализации (δ_4). Только в последней КТ после тех же самых шагов выполняется интерпретация характера проявления неисправности (ε_1); восстановление после сбоя (неявный способ или ζ_1); блокирование отказавшего канала (η_1). Специфика способа восстановления проявляется при выполнении: выбора согласованного значения в ИН, обнаружения неисправности и ее локализации. Механизмы реализации остальных шагов рассмотрены в разделе 3.2.3.1.

Выбор согласованного значения в ИН. Структура ИН должна быть такой же, какая требуется для восстановления с локализацией части "враждебных" и любых "дружественных" форм проявления неисправностей (раздел 3.2.3.3). Выбор согласованного значения осуществляется по большинству совпадающих, содержащихся в главной диагонали B_n. Несмотря на то, что при проявлении "враждебной" неисправности только в некоторых ИН (из всех одновременно сформированных в КПР) в главной диагонали B_n содержится рассогласованный элемент, это обстоятельство не повлияет на результат выбора согласованного значения всеми исправными ЦВМ. Таким образом, механизм выбора согласованного значения в основном совпадает с β_1 (раздел 3.2.3.1).

Обнаружение неисправности. Осуществляется механизмом γ_5 в два этапа. Сначала в каждой ЦВМ независимо от остальных формируется значение признака потенциального проявления неисправности по наличию/отсутствию хотя бы одного рассогласованного элемента в B_n (так же, как в γ_1). Затем осуществляется обмен между всеми ЦВМ (с использованием алгоритма ВИС) значениями этих признаков и формируется вспомогательный набор – вектор D_n размерности N. Любой i-й элемент D_n ($i = \overline{1,N}$) содержит значение признака потенциального проявления неисправности, сформированного в i-й и переданного в n-ю ЦВМ. В том случае, если значение хотя бы одного элемента D_n соответствует фактическому проявлению неисправности, начинается восстановление одновременно всеми исправными ЦВМ, в противном случае обработка КТ завершается. Таким образом, использование D_n обеспечивает одновременность начала восстановления всеми исправными ЦВМ в КПР. Если же пытаться начинать восстановление только на основании рассогласованного элемента в ИН, то при некоторых формах проявления "враждебной" неисправности исправные ЦВМ обнаружат ее неодновременно, что нарушит управление ресурсами комплекса. Дело в том, что если неисправность проявилась только во втором раунде обмена при формировании B_n, то рассогласованные элементы могут содержаться не во всех ИН, одновременно сформированных в комплексе. Условия, требуемые для реализации γ_5: минимально необходимая кратность резервирования ЦВМ равна четырем; режим внутрикомплексного обмена – синхронный.

Локализация неисправного канала. Основана на закономерном характере мест расположения рассогласованных элементов во всех B_n, одновременно сформированных исправными ЦВМ комплекса. Пусть f – номер ЦВМ, неисправность которой проявляется во "враждебной" форме. Рассогласованные элементы могут содержаться только в f-й строке и f-м столбце B_n. Во всех B_n, одновременно сформированных в комплексе, число рассогласованных элементов в f-й строке одно и то же [44]. По характеру расположения рассогласованных элементов в B_n, позволяющему всем или только некоторым исправным ЦВМ сформировать однозначный результат локализации, множество всех возможных "враждебных" форм проявления неисправности было разделено на три группы. Первая группа содержит формы, позволяющие всем

исправным ЦВМ однозначно локализовать неисправную в момент обнаружения неисправности. При этом в f-й строке должно быть не менее двух рассогласованных элементов. Вторая группа состоит из форм, в которых однозначный результат локализации формируется постепенно в нескольких КТ, за счет искусственной генерации признаков безуспешного завершения внутрикомплексного обмена между исправными и неисправной ЦВМ. Группа ЦВМ, сформировавших в некоторой i-й КТ (нумерация с момента обнаружения неисправности) однозначный результат локализации, начиная с $(i+1)$-й, искусственно генерирует значения признаков безуспешного завершения обмена с неисправной ЦВМ, рассылаемых остальным исправным во втором раунде формирования B_n. Тем самым у остальных исправных ЦВМ увеличивается число рассогласованных элементов в B_n, позволяя некоторым (или всем) однозначно локализовать неисправную в $(i+1)$-й или последующих КТ. По истечении не более трех КТ с момента проявления неисправности либо все ЦВМ однозначно локализуют неисправную, либо большинством исправных ЦВМ (кроме, может быть, одной) будет сформирован неоднозначный результат локализации [44]. Под неоднозначным понимается результат локализации в виде двух несовпадающих значений, лишь одно из которых является правильным. Последняя, третья группа форм проявления неисправности позволяет локализовать неисправную ЦВМ только за счет мажоритарного выбора по большинству совпадающих значений неоднозначных результатов локализации, содержащихся во вспомогательных наборах E_n и F_n (табл. 5). E_n и F_n являются векторами размерности N, n-й элемент которых содержит первое и второе значения неоднозначного результата локализации, определенного в n-й ЦВМ. Обозначим рассмотренный механизм локализации как δ_4. Условия, необходимые для его реализации: минимально необходимая кратность резервирования ЦВМ равна четырем; режим внутрикомплексного обмена – синхронный; режим управления ресурсами комплекса – ПВ.

3.2.4. *Механизмы ядра, реализующие управление ресурсами комплекса.* Они обеспечивают поддержку механизмам гипервизора, выполняющим локализацию неисправного устройства в нескольких КТ с момента проявления неисправности. Необходимость выполнения локализации именно в нескольких КТ вызвана следующими причинами: число элементов ИН, содержащих "правильные" значения в момент проявления неисправности, недостаточно велико для выбора согласованного значения мажоритарным способом (δ_2); формы проявления "дружественных" неисправностей в момент их обнаружения не позволяют различить, какое устройство в канале неисправно – ЦВМ или шина (δ_3); формы проявления "враждебных" неисправностей не дают достаточного числа рассогласованных элементов в ИН для однозначного определения неисправной ЦВМ всеми исправными в момент обнаружения неисправности (δ_4). При восстановлении ВП накапливается дополнительная информация о формах проявления неисправности, получаемая за счет коррекции характера управления ресурсами комплекса по сравнению с используемым в стационарном ВП. Она затрагивает работу супервизора и средств внутрикомплексного обмена. Обозначим механизмы ядра, осуществляющие коррекцию, символами μ_k, где k – порядковый номер механизма. Функции супервизора и средств внутрикомплексного обмена, корректируемые механизмами ядра, приведены на рис. 7.

Для того, чтобы поддержать работу δ_2, супервизору требуется организовать возврат к предыстории развития ВП до момента возникновения неисправности, а средствам обмена данными между КПР и абонентами во внешней среде МВС временно блокировать обмен при восстановлении ВП. Возврат к предыстории осуществляется с помощью μ_1 – записи/считывания значений параметров состояния ВП в КТ, временная блокировка обмена с абонентами – с помощью μ_2. Механизмы μ_1 и μ_2 реализуют режим ВН управления ресурсами комплекса. В условиях стационарного ВП в каждой КТ μ_1 осуществляет запись в память ЦВМ значений параметров состояния ВП (в точках хранения) и согласованных значений результатов выполнения

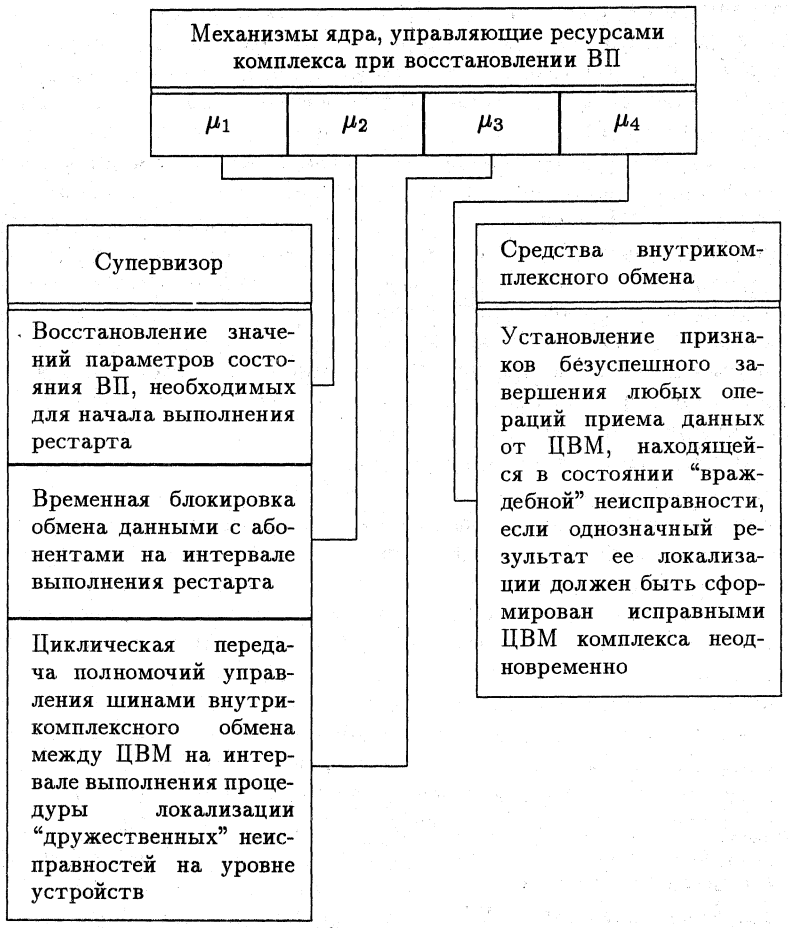

Рис. 7. Элементы ОС, корректируемые механизмами ядра

прикладных задач (в точках сравнения). В момент проявления неисправности μ_1 восстанавливает значения параметров состояния ВП, соответствующие началу выполнения процедуры рестарта (рис. 4). При прохождении каждой точки сравнения на интервале восстановления этот же механизм восстанавливает эталонные значения, необходимые для контроля на совпадение со значениями результатов повторно выполненных прикладных задач. При повторном выполнении прикладных задач необходимо осуществлять временную блокировку обмена с абонентами, поскольку повторное обращение к ним с одним и тем же запросом может нарушить логику работы системы управления. Механизм μ_2 временной блокировки обмена работает следующим образом [33, 45]. Счетчик обмена устанавливается в нулевое значение при входе в каждую точку хранения и получает приращение после каждой успешно выполненной точки сравнения, содержащейся в операции обмена с абонентами. В момент проявления неисправности текущее значение счетчика присваивается системной переменной, устанавливающей порог блокировки обмена. При повторном выполнении прикладных задач от точки хранения обмен комплекса с внешней средой блокирован до тех пор, пока значение счетчика не превысит порогового, установленного в этой переменной, после чего обмен деблокируется.

Механизмы μ_3 и μ_4 реализуют управление ресурсами комплекса в режиме ПВ. Для поддержки δ_3 используется механизм μ_3 – циклической передачи полномочий

управления шинами внутрикомплексного обмена между ЦВМ. В стационарном ВП полномочия управления шинами могут быть постоянным образом распределены между ЦВМ (*FTMP*) или циклически перераспределяться в каждом МЦ (*SIFT*) [36, 39]. При восстановлении ВП должно осуществляться циклическое перераспределение полномочий управления шинами, чтобы одна и та же неисправность проявилась в нескольких операциях внутрикомплексного обмена, осуществленных различными передающими и принимающими данные ЦВМ. Работу δ_4 поддерживает μ_4, осуществляющий в каждой ЦВМ связь между средствами внутрикомплексного обмена и гипервизором. Механизм μ_4 устанавливает значения признака безуспешного приема сообщений от неисправной ЦВМ в исправную, сформировавшую однозначный результат локализации, не являющийся общим для всего комплекса (что распознается автоматически по характеру расположения рассогласованных элементов в B_n). В стационарном ВП, а также при всех иных вариантах формирования результата локализации гипервизор принимает от средств внутрикомплексного обмена фактические значения признаков завершения обмена.

3.2.5. *Сравнительный анализ отказоустойчивости КПР, потенциально достижимой при различных способах управления ВП.* Перечислим шаги обработки КТ, которые позволяют использовать различные механизмы гипервизора в зависимости от способа управления ВП: выбор согласованного значения в ИН; локализация неисправности; восстановление после сбоев; блокирование отказавшей ЦВМ. Механизмы гипервизора, реализующие один и тот же шаг, позволяют упорядочить их по убыванию степени потенциально достижимой отказоустойчивости, определяемой на качественном уровне. Для разных шагов могут быть применены различные параметры оценки качества реализации отказоустойчивости КПР. *Механизм гипервизора, использование которого оценивается наибольшей степенью отказоустойчивости, назовем предпочтительным среди всех, применимых для реализации конкретного шага обработки КТ.* При проведении сравнительного анализа предполагается, что ресурсы достаточны для использования любого механизма. Целью анализа является определение способов управления ВП в КПР, позволяющих выполнить наибольшее число шагов обработки КТ с помощью предпочтительных механизмов.

Анализ восстановления ВП в КПР при "дружественных" формах проявления неисправностей. Для определения предпочтительных механизмов гипервизора используем следующие параметры отказоустойчивости КПР.

а. Возможность реализации защиты от отказа, который может возникнуть при попытке восстановления либо вследствие неполноты тестовой проверки значений элементов ИН (β_3), либо неправильной интерпретации в восстанавливаемой ЦВМ контроля на совпадение собственного результата вычислений с эталонным значением (δ_2) (с помощью β_1 или β_2).

б. Возможность раздельной локализации неисправных ЦВМ и шин внутрикомплексного обмена (с помощью δ_3).

в. Возможность восстановления информации в области хранения как программ, так и данных после сбоев ЦВМ (с помощью ζ_1).

г. Возможность реализации защиты от отказа, который может возникнуть при попытке восстановления вследствие невыполнения механизма самоблокировки в отказавшей ЦВМ (с помощью η_1).

Механизмы β_1 или β_2, предпочтительные по параметру *а*, реализуемы в тех КПР, в которых кратность резервирования ЦВМ не менее трех. В дублированном КПР результат восстановления зависит либо от полноты тестовых проверок значений элементов ИН, либо от успешного/безуспешного завершения рестарта в восстанавливаемой ЦВМ. Механизм δ_3, предпочтительный среди остальных механизмов локализации неисправности по параметру *б*, применим только при следующих характеристиках структуры КПР и управления ВП: кратность резервирования ЦВМ должна иметь значение не менее трех; режим внутрикомплексного обмена должен

Таблица 7

Оценка реализации предпочтительных механизмов гипервизора, обеспечивающих восстановление ВП при "дружественных" формах проявления неисправностей в КПР

Возможность повышения качества отказоустойчивости КПР с помощью предпочтительных механизмов гипервизора	Кратность резервирования ЦВМ и способы управления ВП						
	Режим внутрикомплексного обмена	Асинхронный		Синхронный			
	Минимально необходимая кратность резервирования ЦВМ	2	3	2		3	
	Режим управления ресурсами комплекса при восстановлении ВП	–	–	–	ВН	–	ПВ
а		–	+	–	–	+	+
б		–	–	–	–	–	+
в		–	–	+	+	+	+
г		–	–	–	–	+	+

быть синхронным; при восстановлении необходимо реализовать управление ресурсами комплекса в режиме ПВ. Механизм δ_3 позволяет локализовать неисправность на уровне устройств, любые другие механизмы – только на уровне каналов. Предпочтительный по параметру *в* механизм ζ_1 может быть реализован только в синхронном КПР. Любые другие способы восстановления после сбоев ЦВМ позволяют восстановить информацию только в области хранения данных. Наконец, предпочтительный по параметру *г* механизм η_1 блокирования отказавшей ЦВМ может быть реализован в синхронном КПР с кратностью резервирования не менее трех. В асинхронном КПР при той же кратности резервирования защита от несрабатывания самоблокировки в отказавшей ЦВМ может быть обеспечена только при использовании стратегии маскирования. Защиту дублированного КПР от данного проявления отказа реализовать невозможно. Итоговые данные по реализуемости предпочтительных механизмов гипервизора приведены в табл. 7.

Все предпочтительные механизмы реализуемы в синхронном КПР с кратностью резервирования ЦВМ не менее трех, с управлением ресурсами комплекса в режиме ПВ. Почти все предпочтительные механизмы (кроме δ_3) реализуемы в синхронном КПР с кратностью резервирования не менее трех. Ни одного предпочтительного механизма невозможно реализовать в асинхронном дублированном КПР. При остальных способах организации ВП реализуем только один механизм.

Анализ восстановления ВП в КПР при "дружественных" и "враждебных" формах проявления неисправностей. Известные способы восстановления ВП различаются только по объему форм проявления "враждебных" неисправностей, которые могут быть успешно локализованы. Этот признак и может быть использован для оценки отказоустойчивости КПР (предпочтительным является механизм δ_4). Оба способа восстановления реализуемы в синхронном КПР с кратностью резервирования ЦВМ не менее четырех. Для реализации предпочтительного механизма необходимо, кроме того, управление ресурсами комплекса в режиме ПВ.

Сформулируем закономерности, связывающие качество реализации отказоустойчивости статического КПР с характеристиками структуры и способами управления ВП. Из сравнительного анализа следует, что сочетание синхронного режима внутри-

Таблица 8

Функции элементов ОС, реализующие управление ВП в статическом КРЗ

Элементы ОС	Функции элементов ОС, выполняемые в стационарном и нестационарном процессах
Супервизор	В основной ЦВМ – диспетчеризация по времени прикладных, системных задач, обмена данными с резервными ЦВМ и с абонентами во внешней среде. В резервных ЦВМ – диспетчеризация по времени системных задач и внутрикомплексного обмена
Ядро	Реализация синхронного режима выполнения внутрикомплексного обмена. Восстановление значений параметров состояния ВП в момент начала выполнения процедуры рестарта и временная блокировка обмена с абонентами до ее завершения (только в нестационарном процессе)

комплексного обмена и достаточно высокой кратности резервирования ЦВМ даже без использования управления ресурсами комплекса создает предпосылки для существенного повышения отказоустойчивости КПР: позволяет надежно осуществлять выбор согласованных значений в ИН, локализовать хотя бы часть "враждебных" форм проявления неисправностей, восстанавливать после сбоев область хранения программ в ОЗУ и надежно блокировать отказавшую ЦВМ по сигналам от всех исправных. Управление же ресурсами комплекса позволяет повысить при этом качество локализации – при любых "враждебных" формах проявления успешно локализовать неисправность на уровне каналов, а при "дружественных" – на уровне устройств.

Сформулируем правило выбора предпочтительного режима внутрикомплексного обмена, исходя из наиболее вероятной причины возникновения неисправностей. Временной механизм диспетчеризации и синхронный режим внутрикомплексного обмена создают необходимые предпосылки для управления ресурсами комплекса при восстановлении ВП. Благодаря этой поддержке гипервизора ядром имеется возможность реализовать сложные способы восстановления в синхронном КПР, требующие последовательной обработки нескольких КТ с момента проявления неисправности. В асинхронном КПР механизмы гипервизора реализуемы без поддержки ядром, а потому способы восстановления являются более простыми, реализуемыми при обработке одной КТ. Единственным существенным недостатком способов восстановления в синхронном КПР по сравнению с асинхронным является незащищенность комплекса от неисправностей, вызванных неблагоприятным воздействием окружающей среды, одновременно проявляющихся во всех ЦВМ (например, сбой при электромагнитных возмущениях). Таким образом, если вероятность неблагоприятного воздействия окружающей среды на аппаратуру МВС соизмерима с вероятностью возникновения физических неисправностей, то предпочтительно использовать асинхронный режим обмена.

Из анализа условий, необходимых для реализации механизмов гипервизора (табл. 6, 7), следует вывод о том, что возможность применения любого способа восстановления в статическом КПР ограничена следующими параметрами: минимально необходимой кратностью резервирования ЦВМ, режимами внутрикомплексного обмена (синхронным или асинхронным) и управления ресурсами комплекса (ПВ или ВН). Последние характеризуют способы управления ВП. Каждому способу восстановления должна соответствовать собственная концептуальная модель, объясняющая поддержку гипервизора ядром. Таким образом, в базовом подклассе МВС с детерминированным распределением ресурсов связь средств управления и восста-

Таблица 9

Функции элементов ОС, реализующие восстановление ВП в статическом КРЗ

Элементы ОС	Функции элементов ОС	
	Стационарный процесс	Нестационарный процесс
Ядро	Контроль допустимого времени ожидания приема сообщений-откликов от всех ЦВМ комплекса. Запись параметров состояния ВП по завершении процедуры обработки КТ, обеспечивающих возможность повторного выполнения прикладных задач при восстановлении ВП с помощью процедуры рестарта	Обнаружение неисправности по признаку аномального завершения попытки синхронизации
Гипервизор	–	Реализация локализации неисправности и замещения отказавшей ЦВМ
Средства межмашинного обмена	Установление признаков успешного завершения внутрикомплексного обмена	Установление признаков безуспешного завершения обмена с неисправной ЦВМ. Восстановление информации после сбоев шины
Средства встроенного контроля	Подтверждение исправного состояния ЦВМ при выполнении диагностических программ	Обнаружение признаков проявления отказа ЦВМ и выполнение ее самоблокировки

новления ВП описывается несколькими концептуальными моделями, число которых равно числу способов восстановления.

3.3. *Реализация основных элементов ОС в статическом КРЗ.* Единственный способ управления ВП, реализованный в статическом КРЗ, имел следующие характеристики. Прикладные задачи выполнялись только в основной ЦВМ, которая в синхронном режиме в каждой КТ передавала исходные данные в резервные ЦВМ. Для диспетчеризации внутрикомплексного обмена был использован временной механизм (раздел 3.2.1). Способ восстановления имел существенное сходство с вариантом, реализованным в синхронном дублированном КПР, использующем рестарт (раздел 3.2.3.1). Функции элементов ОС статического КРЗ приведены в табл. 8 и 9.

Из таблиц видно, что в стационарном ВП обработка КТ выполняется только вспомогательными элементами ОС, а гипервизор используется в нестационарном процессе. Эта специфика способа восстановления ВП в КРЗ обусловлена нерезервированным характером вычислений единственной активной копии прикладной задачи и отсутствием ИН в обработке КТ. Обнаружение и локализация неисправности выполняются только с помощью тестовых проверок, характер которых различен для основной и резервных ЦВМ. Термин "результат локализации" применительно к способу восстановления в данном случае означает интерпретацию сочетания значений признаков обнаружения неисправности, сформированных средствами внутрикомплексного обмена и встроенного контроля ЦВМ. Проведем анализ способа восстановления ВП в КРЗ. Затем сравним способы восстановления в КРЗ и синхронном дублированном КПР.

3.3.1. *Реализация восстановления ВП в КРЗ.* Обработка КТ в КРЗ состоит из шагов: обнаружение неисправности, локализация неисправной ЦВМ, реконфигурация комплекса. В любой ЦВМ отсутствует возможность различать характер проявления неисправности другой ЦВМ, обнаруживаемой по признаку безуспешно-

го завершения внутрикомплексного обмена. Поэтому любое восстановление производится с замещением неисправной ЦВМ. Рассмотрим выполнение каждого шага обработки КТ в основной и резервных ЦВМ [45 – 51].

Обнаружение неисправности. Следует различать два типа механизмов обнаружения неисправности в КРЗ: первый имеет характер самопроверки работоспособности ЦВМ, второй – контроля исправности любой другой ЦВМ комплекса при внутрикомплексном обмене. Самопроверка работоспособности осуществляется с помощью встроенного контроля и тестовых программ, работающих независимо от гипервизора и внутрикомплексного обмена. В основной ЦВМ тестовые программы могут оценивать принадлежность результата выполнения прикладной задачи области допустимых значений (так же, как β_3, раздел 3.2.3.1). В резервных ЦВМ могут быть использованы только тесты проверки работоспособности аппаратуры. Во всех известных КРЗ основная и резервные ЦВМ периодически обменивались сигналами успешного/безуспешного выполнения самопроверки, причем этот обмен выполнялся одновременно с передачей данных от основной к резервным ЦВМ в начале выполнения каждой прикладной задачи. Контроль работоспособности ЦВМ, являющейся партнером в обмене, выполнялся средствами внутрикомплексного обмена. Нарушение формата передачи данных, так же как отсутствие приема сообщения в заданном интервале времени ожидания, было интерпретировано как признак проявления неисправности.

Локализация неисправной ЦВМ. Работа механизма локализации была основана на предположениях об одиночном характере проявления неисправности и о более высокой надежности работы средств контроля по сравнению с контролируемой ими аппаратурой. Поэтому наиболее вероятными предполагались следующие сочетания результатов тестовой самопроверки ЦВМ и признака завершения внутрикомплексного обмена, по которым неисправность одновременно должна быть обнаружена и локализована. Либо тест самопроверки выполнен успешно и обнаружен признак безуспешного завершения обмена, что требовалось интерпретировать как неисправность ЦВМ – партнера в обмене. Либо безуспешно выполнен тест самопроверки, что независимо от успешного/безуспешного завершения обмена должно означать проявление неисправности самой ЦВМ. В тех системах, где от правильности определения неисправной ЦВМ зависела безопасность функционирования объекта, локализация выполнялась вручную [52].

Реконфигурация комплекса осуществлялась путем замещения неисправной ЦВМ. Если была неисправна основная ЦВМ, то одна из резервных принимала на себя обмен с внешней средой МВС и выполняла рестарт от ближайшей точки хранения. Процедура рестарта в резервной ЦВМ осуществлялась с такой же поддержкой от ядра, как в синхронном дублированном КПР (μ_1 и μ_2, раздел 3.2.4). Неисправная основная ЦВМ должна была выполнить самоблокировку. В случае неисправности резервной ЦВМ для осуществления реконфигурации комплекса требовалось только выполнить ее самоблокировку.

3.3.2. *Сравнительный анализ восстановления ВП в КРЗ и в синхронном дублированном КПР.* В них восстановление после неисправностей, интерпретированных как отказы, осуществляется с помощью рестарта, реализуемого одинаковыми механизмами. Таким образом, имеет смысл сравнивать только механизмы обнаружения и локализации неисправности. Способ контроля на совпадение между двумя значениями элементов ИН в КПР (γ_2) обладает большей вероятностью успешного обнаружения неисправности, чем тестовая самопроверка результата выполнения одной активной копии задачи в КРЗ, причем различие проявляется более значительно в системах жесткого РВ, где ограничены ресурсы времени на выполнение тестовой программы [53]. Таким образом, синхронный дублированный КПР обладает существенно меньшей вероятностью отказа, вследствие безуспешной попытки обнаружения одиночной неисправности, чем КРЗ.

Таблица 10

Функции основных элементов ОС, реализующие управление и восстановление ВП при выполнении межкомплексного обмена

Элементы ОС	Функции элементов ОС		
	Управление ВП в стационарном и нестационарном процессах	Восстановление ВП	
		Стационарный процесс	Нестационарный процесс
Супервизор	Диспетчеризация межкомплексного обмена данными по времени	–	–
Ядро	Реализация синхронного режима межкомплексного обмена	Контроль допустимого времени ожидания приема сообщений-откликов от всех ЦВМ, участвующих в обмене данными	Обнаружение неисправности по признаку аномального завершения попытки межкомплексной синхронизации
Гипервизор	–	В принимающем комплексе осуществление выбора согласованного значения в ИН. В передающем и принимающем комплексах контроль признаков успешного завершения межкомплексного обмена	В передающем и принимающем данные комплексах реализация способа восстановления ВП с локализацией "дружественных" форм проявления неисправностей

Ограничения реализации функций супервизора и ядра в статическом КРЗ, обусловившие использование единственно возможного способа управления ВП, вызваны необходимостью обеспечить синхронное выполнение тестовых проверок в основной и резервных ЦВМ в стационарном процессе и реализовать процедуру рестарта при восстановлении. Концептуальная модель, описывающая связь управления и восстановления ВП в статическом КРЗ, естественно, должна отличаться от любой модели, применимой для статического КПР.

3.4. *Реализация основных элементов ОС в статических МВС из нескольких синхронных КПР.* Для всех МВС с детерминированным распределением ресурсов характерно жесткое ограничение на затраты времени при выполнении межкомплексного обмена. Именно это ограничение вынуждает формировать систему только из синхронных КПР и не использовать рестарт для восстановления ВП, хотя последнее не является существенным. Примерами практической реализации статических МВС из нескольких синхронных КПР являются экспериментальная система *AIPS* [54 – 58], а также МВС космического корабля "*Space Shuttle*" (на орбитальном этапе полета) [52]. В этих системах разные комплексы выполняли циклические прикладные задачи с кратными периодами выдачи решения. Функции ОС выполнялись в каждом комплексе независимо от остальных. Связь между ними проявлялась только в операциях межкомплексного обмена. Поэтому анализ основных элементов ОС состоял в исследовании способов восстановления ВП в случае проявления неисправности при его выполнении. Функции основных элементов ОС статических МВС из нескольких синхронных КПР приведены в табл. 10.

Таблица 11

Функции основных элементов ОС, реализующие управление ВП в динамической МВС из нескольких синхронных КПР

Элементы ОС	Функции элементов ОС, выполняемые в стационарном и нестационарном процессах
Локальный супервизор	Диспетчеризация прикладных задач и внутрикомплексного обмена в каждом КПР независимо от остальных
Глобальный супервизор	Формирование КПР и диспетчеризация межкомплексного обмена по времени
Ядро	Реализация синхронного режима межмашинного обмена, обеспечение одновременного формирования всех КПР в МВС

Реализация восстановления ВП в случае проявления неисправности при выполнении межкомплексного обмена. В каждой операции межкомплексного обмена данными гипервизоры передающего и принимающего комплексов выполняли обработку КТ. Для передающего комплекса элементами ИН являлись значения признака успешного/безуспешного завершения межкомплексного обмена, для принимающего – значения передаваемых данных или признака безуспешного завершения обмена. Все исправные ЦВМ принимающего комплекса должны были выбирать одни и те же согласованные значения данных. Гипервизоры в обоих комплексах (как передающем, так и принимающем) имели возможность реконфигурации межкомплексных связей после отказов ЦВМ в любом КПР или шин. Для обработки КТ в межкомплексном обмене были использованы механизмы гипервизора, которые обеспечивали в синхронном КПР восстановление с локализацией "дружественных" форм проявления неисправностей. В случае сбоев при приеме данных требовалось восстановить только содержание области хранения данных в ОЗУ той ЦВМ, которая не смогла принять их по своей шине. Реконфигурация межкомплексных связей осуществлялась следующим образом. Если произошел отказ ЦВМ в передающем КПР, требовалось перераспределить полномочия доступа к шинам межкомплексного обмена между исправными ЦВМ. Если произошел отказ ЦВМ в принимающем КПР или отказала шина, требовалось установить запрет на обращение к отказавшей части системы в межкомплексном обмене.

Средства обеспечения отказоустойчивости статических МВС из нескольких синхронных КПР реализуют несколько различных способов восстановления (при проявлении неисправностей во внутрикомплексном и межкомплексном обмене). Каждый способ восстановления описывается собственной концептуальной моделью.

3.5. *Реализация основных элементов ОС в динамических МВС из нескольких синхронных КПР.* Реализация динамических систем имеет пока чисто экспериментальный характер. По публикациям известно порядка десяти динамических МВС [59 – 61]. Наиболее подробное описание структуры ОС и механизмов обеспечения отказоустойчивости приведено по системам: *SIFT* [62 – 71], *FTMP* [35, 72], *CRMMFCS* [73 – 75], *MAFT* [76 – 80], μC^* (трехверсионное программирование), а также МВС самолета *HiMAT* (двухверсионное программирование) [13 – 16, 81]. *Функцию ОС динамической МВС, реализующую в РВ изменение числа комплексов и принадлежности к ним любой исправной ЦВМ, назовем формированием КПР.* Способы управления ВП в динамических МВС различаются реализацией формирования КПР. Преимущества каждого способа над остальными (проявляющиеся либо в числе ЦВМ, либо в затратах процессорного времени на выполнение прикладных задач) имеют место только при определенных характеристиках прикладных задач.

Таблица 12

Функции основных элементов ОС, реализующие восстановление ВП в динамической МВС из нескольких синхронных КПР

Элементы ОС	Функции элементов ОС	
	Стационарный процесс	Нестационарный процесс
Ядро	Контроль допустимого времени ожидания приема сообщений-откликов от всех ЦВМ, участвующих во внутрикомплексной (межкомплексной) синхронизации	Обнаружение неисправности по признаку аномального завершения попытки внутрикомплексной (межкомплексной) синхронизации. Циклическая передача полномочий управления шинами межкомплексного обмена между ЦВМ при реализации восстановления ВП на уровне МВС с локализацией "дружественных" неисправностей на уровне устройств
Локальный гипервизор	В каждом КПР выполнение шагов обработки КТ, реализующих выбор согласованного значения и контролирующих идентичность результатов решения активных копий прикладных задач	Реализация восстановления ВП в КПР в случае проявления неисправностей при выполнении внутрикомплексного обмена
Глобальный гипервизор	Контроль признаков успешного завершения межкомплексного обмена	Реализация восстановления ВП в МВС в случае проявления неисправностей при выполнении межкомплексного обмена

Способы восстановления ВП после неисправностей ЦВМ (шин) во всех известных нам системах одинаковы (различны). *Функции каждого основного элемента ОС любой динамической МВС, в зависимости от того, выполняются они только в КПР или во всей системе, называют соответственно локальными или глобальными.* В каждой конкретной МВС могут быть реализованы либо оба типа функций (локальные и глобальные) основных элементов ОС, либо только один из них. При анализе реализации основных элементов ОС нас будут интересовать прежде всего специфичные механизмы выполнения глобальных функций. Функции основных элементов ОС динамических МВС приведены в табл. 11 и 12.

Анализ состоит из двух шагов: анализа способов формирования КПР (эту функцию выполняет глобальный супервизор); анализа способов восстановления ВП, реализуемых локальным и глобальным гипервизорами.

3.5.1. *Анализ способов формирования КПР.* КПР в динамических МВС используются как для выполнения прикладных задач и обработки КТ по их результатам (когда одновременно работают несколько КПР), так и для восстановления ВП после проявления неисправностей в межкомплексном обмене (когда из всех исправных ЦВМ формируется единый КПР). Формирование КПР осуществляется одновременно во всех ЦВМ с помощью временного механизма диспетчеризации, единого для всей системы. При этом каждая ЦВМ имеет в памяти таблицу распределения задач между всеми остальными исправными ЦВМ и благодаря ее содержанию распознает, с какими из них она образует КПР при выполнении каждой задачи.

Таблица 13

Способы формирования КПР в динамических МВС

Номер способа	Интервал времени работы	Одинаковая/различная кратность резервирования копий прикладных задач	Форма резервирования активных копий прикладных задач	Примеры систем, в которых реализован способ формирования КПР
1	такт, достаточный для одной задачи	одинаковая	пространственная	–
2	–"–	–"–	пространственно-временная	*CRMMFCS*
3	–"–	различная	пространственная	*SIFT*, *MAFT*
4	–"–	–"–	пространственно-временная	–
5	МЦ	одинаковая	пространственная	*FTMP*
6	–"–	–"–	пространственно-временная	–
7	–"–	различная	пространственная	–
8	–"–	–"–	пространственно-временная	–

Будем различать способы формирования КПР по трем параметрам, каждый из которых имеет два значения: величина времени работы совпадает либо с тактом выполнения одной прикладной задачи, либо с величиной МЦ; кратность резервирования активных копий – одинаковая или различная для всех задач; форма резервирования активных копий прикладных задач, выполняемых в синхронном КПР, – пространственная или пространственно-временная. Последний из перечисленных параметров требует пояснений, поскольку он специфичен только для динамических МВС с детерминированным распределением ресурсов. В статических МВС используется только пространственная форма резервирования активных копий, которая означает их одновременное выполнение разными ЦВМ. Пространственно-временная форма резервирования означает поочередное выполнение активных копий задач разными ЦВМ системы в разных интервалах времени. Поясним ее на примере управления ВП в *CRMMFCS* [73 – 75]. В этой системе интервал работы комплекса совпадает с тактом выполнения прикладной задачи; кратность резервирования копий для всех задач одна и та же, равная трем. Пусть требуется в одном и том же интервале выполнить K независимых задач, одновременно на разных ЦВМ. Число задач должно быть больше кратности резервирования копий. Для их выполнения в *CRMMFCS* используются только K ЦВМ, которые выполняют один и тот же набор задач в течение трех тактов. На каждом такте в МВС выполняется только одна активная копия любой задачи, однако на трех последовательных тактах выполняются три копии этой задачи тремя разными ЦВМ. Четвертый такт предназначен для обработки КТ одновременно всеми ЦВМ. Каждая ЦВМ обменивается результатами вычислений только с теми из остальных, которые образовывали с ней КПР (выполняли активные копии задач). Нетрудно представить систему, состоящую из нескольких пространственно-временных КПР, отличающуюся от *CRMMFCS* по способу формирования комплексов (например, с интервалом работы КПР, совпадающим с величиной МЦ). Локальный супервизор реализован в тех МВС, в которых интервал

времени работы комплекса равен МЦ (за это время КПР выполняет несколько прикладных задач). Из практически реализованных МВС он используется только в *FTMP*. Глобальный супервизор реализован в ОС всех динамических МВС, именно он осуществляет формирование КПР. По приведенным выше параметрам различимы 8 способов формирования КПР, причем в известных нам ОС практически были реализованы только три из них (табл. 13).

Любой способ формирования КПР должен минимизировать затраты ресурсов и позволяет эффективно управлять ВП только в том случае, если набор задач обладает требуемыми характеристиками. Не реализованными оказались те способы, которые предъявляют к характеристикам задач требования, вероятность удовлетворения которых в реальных системах чрезвычайно мала, или требуют чрезмерно сложных алгоритмов управления ВП.

3.5.2. *Способы восстановления ВП, реализуемые локальным и глобальным гипервизорами.* В любой динамической МВС локальный гипервизор используется для обработки КТ по результатам решения активных копий прикладных задач. В динамических МВС локальный гипервизор реализует способ восстановления с локализацией "дружественных" неисправностей на уровне каналов (с помощью механизмов, применимых в статическом синхронном КПР с кратностью резервирования ЦВМ не менее трех). Восстановление после сбоев ЦВМ выполняется, как правило, неявным способом. Механизм ζ_1, реализующий явный способ восстановления после сбоев, применяется только в *FTMP*. Глобальный гипервизор выполняет восстановление в случае проявления неисправностей при выполнении межкомплексного обмена. Этот элемент ОС реализован только в тех системах, в которых резервированы общие шины, используемые как для внутри- , так и межкомплексного обмена (*SIFT*, *MAFT*, *FTMP*). В *CRMMFCS* любые операции межмашинного обмена выполняются по нерезервированной шине, поэтому в ее ОС функции глобального гипервизора не реализованы. С помощью глобального гипервизора во всех динамических системах реализован способ восстановления с локализацией "дружественных" неисправностей ЦВМ и шин на уровне устройств. Кроме того, в ОС *SIFT* и *MAFT* реализован алгоритм ВИС, используемый для согласования работы таймеров во всех ЦВМ.

Любой способ восстановления в динамической МВС, реализуемый как локальным, так и глобальным гипервизорами, совпадает с каким-либо из применяемых в статическом синхронном КПР. Таким образом, все способы восстановления в динамических МВС могут быть описаны концептуальными моделями, которые применимы для статического синхронного КПР.

3.6. *Сравнительный анализ эффективности способов управления и восстановления ВП в статических и динамических МВС из нескольких КПР.* Чтобы определить область возможной реализации статических и динамических систем, рассмотрим соотношение между потребностями в затратах для выполнения всех прикладных задач в МВС и ресурсами одной ЦВМ. Затем определим преимущества способа обеспечения отказоустойчивости динамических МВС, а также "плату" за реализацию этих преимуществ в виде дополнительных (отсутствующих в статических системах) требований к прикладным задачам и способам реализации основных элементов ОС.

3.6.1. *Ограничения ресурсов, определяющие область возможной реализации статических и динамических МВС.* Соотношение между затратами на выполнение всех прикладных задач в МВС и ресурсами одной ЦВМ, определяющее область возможной реализации МВС с детерминированным распределением ресурсов, приведено на качественном уровне в табл. 14.

Поясним условия возможной реализации подклассов МВС. Необходимым условием возможной реализации динамической МВС является достаточность памяти одной ЦВМ для размещения программ ОС и всех прикладных задач. Если же памяти одной ЦВМ недостаточно, то допустима реализация только статической МВС из

Таблица 14

Область возможной реализации МВС с детерминированным распределением ресурсов, определяемая соотношением между потребностями в затратах процессорного времени ($r_в$) и памяти ($r_п$) для выполнения всех прикладных задач в директивные сроки, и ресурсами, предоставляемыми одной ЦВМ ($R_в$ и $R_п$)

Соотношение по памяти	Соотношение по процессорному времени	
	Ресурсы производительности одной ЦВМ достаточны для выполнения всех прикладных задач в директивные сроки $r_в < R_в$	Ресурсы производительности одной ЦВМ недостаточны для выполнения всех прикладных задач в директивные сроки $r_в > R_в$
Ресурсы памяти одной ЦВМ достаточны для хранения программ и данных ОС и всех прикладных задач $r_п < R_п$	Реализуемы: статический КПР или КРЗ; динамическая система из одного или нескольких КПР	Реализуемы: статическая система из нескольких КПР; динамическая система из нескольких КПР
Ресурсы памяти одной ЦВМ недостаточны для хранения программ и данных ОС и всех прикладных задач $r_п > R_п$	Реализуема только статическая система из нескольких КПР	Реализуема только статическая система из нескольких КПР

нескольких КПР. Если ресурсы памяти одной ЦВМ позволяют разместить в ней программы ОС и всех прикладных задач, а ресурсы производительности достаточны для выполнения всех задач в директивные сроки, то допустима реализация либо статического КПР, либо динамической МВС из одного или нескольких КПР. Реализация статического КПР является предпочтительной, так как в нем функции ОС реализуются наиболее простыми способами. Наконец, если ресурсы памяти позволяют разместить программы ОС и всех прикладных задач в памяти одной ЦВМ, а ресурсы процессорного времени недостаточны для их выполнения в одном комплексе, то выбор должен быть осуществлен между статической и динамической системой из нескольких КПР. Именно для этого соотношения между затратами и ресурсами имеется практическая возможность применения результатов сравнительного анализа статических и динамических систем, который проводится далее.

3.6.2. *Анализ достоинств (недостатков) динамических МВС в реализации отказоустойчивости (в реализации ВП).* Преимущества динамических систем над статическими определяются следующими причинами: полной взаимозаменяемостью ЦВМ; для выполнения прикладных задач с заданной кратностью резервирования требуется меньше ЦВМ (достигается только в случае, когда для разных задач требуется разная кратность резервирования); более высокой обеспеченностью ресурсами для локализации "дружественной" неисправности на уровне устройств (δ_3) и восстановления после сбоев путем копирования памяти (ζ_1) (табл. 15).

В динамических МВС созданы более благоприятные условия для реализации отказоустойчивости, чем в статических. "Платой" за эти достоинства являются дополнительные, отсутствующие в статических системах требования к характеристикам прикладных задач и способам реализации основных элементов ОС (табл. 16).

Таблица 15

Способы достижения более высокой отказоустойчивости динамических МВС по сравнению со статическими с использованием меньшего числа ЦВМ

Преимущества динамических систем над статическими	Характеристики управления ВП, за счет которых обеспечивается преимущество динамических МВС над статическими	Недостатки способа восстановления ВП в статических МВС
Перераспределение исправных ЦВМ между комплексами после отказов. Отказ системы возникает, когда число исправных ЦВМ становится недостаточным для выполнения в РВ всех прикладных задач	В памяти каждой ЦВМ хранятся программы всех прикладных и системных задач МВС. Механизм диспетчеризации всех или некоторых задач реализован на уровне МВС. Преимуществом обладает любая динамическая МВС	Комплексы деградируют независимо. Отказ системы возникает, когда какой-либо комплекс переходит в состояние отказа
Число исправных ЦВМ, требуемое для решения прикладных задач с заданной кратностью резервирования активных копий, меньше, чем в статических системах	Комплексы формируются под каждую задачу с заданной кратностью резервирования копий. Преимуществом обладают только МВС с различной кратностью резервирования ЦВМ в разных КПР (системы типа *SIFT*, *MAFT*)	Ресурсы процессорного времени используются тем менее эффективно, чем больше имеется неравномерность распределения прикладных задач между КПР с различной кратностью резервирования ЦВМ
Ресурс времени, выполняемый для восстановления ВП после сбоев с помощью механизма копирования памяти, больше, чем в статических системах	Комплекс ЦВМ, в котором требуется выполнить копирование памяти, исключается из распределения ресурсов под задачи на любой интервал времени, достаточный для восстановления ВП. Преимуществом обладает любая динамическая МВС	Каждый комплекс должен постоянно решать прикладные задачи. Поэтому для восстановления после сбоев может быть выделен интервал времени, не превосходящий части МЦ, не занятой выполнением прикладных задач
Механизм локализации "дружественной" неисправности на уровне устройств может работать до тех пор, пока остаются исправными не менее трех ЦВМ в МВС	В выполнении этого механизма участвуют исправные ЦВМ всей МВС. Преимуществом обладают только МВС, в которых реализованы функции глобального гипервизора	В выполнении этого механизма участвуют исправные ЦВМ только одного КПР. Поэтому при деградации МВС быстрее, чем в динамической системе, число исправных ЦВМ становится недостаточным для его использования

Во всех КПР любой динамической системы должны быть реализованы одинаковые механизмы диспетчеризации прикладных задач (локальный супервизор), идентичные способы восстановления ВП (локальный гипервизор). Моменты формирования КПР и выполнения межкомплексного обмена должны быть синхронизированы во всей системе (ядро и глобальный супервизор). Кроме того, в тех МВС, в кото-

Таблица 16

Требования к реализации ВП в КПР динамических МВС

Характеристики реализации ВП, которые должны совпадать в КПР	В чем должно проявляться совпадение характеристик реализации ВП	Типы динамических МВС (различимые по способу формирования КПР), в которых требуется обеспечить совпадение характеристик
Оценка затрат процессорного времени на решение каждой прикладной задачи	Величина оценки должна быть одинаковой для всех прикладных задач, решаемых в МВС	Требуется только в тех МВС, в которых интервал работы КПР достаточен для решения одной прикладной задачи
Синхронизация моментов формирования КПР	Моменты формирования КПР должны быть жестко синхронизированы по сигналам таймера, общим для всей МВС	Требуется в любой динамической МВС
Идентичность механизмов диспетчеризации прикладных задач	Возможны варианты: временной механизм диспетчеризации, единый для всех КПР;	Требуется только в тех МВС, в которых интервал работы КПР достаточен для решения одной прикладной задачи
	один и тот же, но не обязательно временной, механизм диспетчеризации для всех КПР	Требуется только в тех МВС, в которых интервал работы КПР совпадает с величиной МЦ
Совпадение способов восстановления ВП в КПР, реализуемых локальными гипервизорами	Во всех КПР должны быть реализованы одинаковые способы восстановления ВП	Требуется в любой динамической МВС

рых интервал работы КПР достаточен для выполнения только одной прикладной задачи, все задачи должны иметь примерно одинаковую величину затрат времени на решение. Последнее из перечисленных требований является наиболее трудно выполнимым в конкретных областях применения. Именно по этой причине во всех динамических МВС, в которых интервал работы КПР позволял решить только одну прикладную задачу (типа *SIFT*, *MAFT*, *CRMMFCS*), прикладное ПО было представлено демонстрационными версиями. Система *FTMP*, в которой интервал работы КПР совпадал с величиной МЦ, используемой в существующих системах управления, оказалась пригодной для работы с реальными версиями прикладного ПО [82]. Таким образом, очевидно, что реализация динамических МВС только в виде экспериментальных вызвана определенными трудностями в реализации ОС, требованиями, предъявляемыми к характеристикам ЦВМ и прикладных задач.

3.7. *Реализация основных элементов ОС с использованием исполнительной системы параллельных ЯВУ для управления межкомплексным обменом.* При использовании исполнительной системы любого параллельного ЯВУ предполагается конкурентная борьба между задачами за ресурсы, что характерно для способа управления ВП в МВС со случайным распределением ресурсов. В немногих известных примерах использования исполнительной системы параллельных ЯВУ для управления

межкомплексным обменом в ОС МВС с детерминированным распределением ресурсов способы реализации основных ее элементов были наиболее простыми из спектра реализуемых в динамических МВС. Специфика использования исполнительной системы параллельных ЯВУ для управления межкомплексным обменом проявлялась лишь в виде ограничений спектра вариантов реализации основных элементов ОС. Известны два примера подобных систем: статическая МВС, состоявшая из троированных синхронных КПР (ОС этой МВС, имеющая собственное имя *HMPOS*, реализована на основе исполнительной системы языка *Ada* [83]), и МВС, предназначенная для управления космическим аппаратом по проекту *Mars'*94 [84]. Последняя система состояла из четырех транспьютеров, которые могли работать либо как статический синхронный КПР, либо как динамическая система из нескольких КПР с пространственно-временной формой резервирования, типа *CRMMFCS* (ОС реализована на основе механизмов языка *Occam*).

В обеих системах диспетчеризация прикладных задач была реализована с помощью временного механизма, использовавшего единую службу системного времени для всех ЦВМ. Обмены выполнялись только в жестко синхронном режиме, по сигналам таймера. В параллельных ЯВУ *Ada* и *Occam* имеются собственные механизмы синхронизации параллельных процессов, но они в рассматриваемых МВС не были использованы в качестве основы синхронизации. Полагаем, что отказ от их использования вызван, главным образом, тем, что языковые механизмы более предпочтительны для организации взаимодействия между разными задачами, чем между активными копиями одной и той же задачи.

Средства обеспечения отказоустойчивости реализовывали восстановление с локализацией "дружественных" неисправностей на уровне каналов. Особенность его реализации в ОС *HMPOS* состояла в том, что независимо от числа исправных ЦВМ в КПР для восстановления после сбоев был использован рестарт. При этом точки хранения должны были совпадать с началом решения каждой прикладной задачи, чтобы восстановление после сбоев могло быть осуществлено путем повторного запуска задачи. По-видимому, реализация рестарта в ОС *HMPOS* оказалась проще любых других способов восстановления после сбоев.

Таким образом, использование исполнительной системы параллельных ЯВУ в качестве основы ОС МВС с детерминированным распределением ресурсов не оказало существенного влияния ни на способ управления ВП, ни на его восстановление.

4. МВС со случайным распределением ресурсов

ОС МВС со случайным распределением ресурсов делятся на подклассы по структуре управления межмашинным обменом и по способу реализации синхронного режима (табл. 17).

Базовым является подкласс ОС с распределенным управлением межмашинным обменом и реализацией синхронизации с помощью системных программ. В подклассе могут быть реализованы несколько механизмов синхронизации и способов восстановления ВП. Некоторые способы управления ВП, реализуемые в нем, позволяют минимизировать затраты ресурсов для решения прикладных задач, однако при этом распределение системных программ между ЦВМ не позволяет осуществить их резервирование на том уровне, который необходим для работы программных средств обеспечения отказоустойчивости. Остальные подклассы отличаются от базового тем, что возможности реализации синхронизации межмашинного обмена и восстановления ВП в них ограничены либо вследствие централизованного управления обменом, либо по причине специфики реализации синхронизации параллельных процессов в конкретных ЯВУ. Отказоустойчивость в них реализована только в ОС с распределенным управлением параллельными процессами и языковой реализацией синхронизации, причем только одним способом. Определим программные объекты, реализующие функции основных элементов ОС.

Таблица 17

Возможности реализации синхронизации межмашинного обмена и восстановления ВП

Способ реализации синхронизации межмашинного обмена	Структура управления машинным обменом	
	Централизованное управление	Распределенное управление
Реализация с помощью только системных программ ОС	ОС, в которых возможности реализации синхронного режима ограничены вследствие централизованного управления межмашинным обменом. Отказоустойчивость МВС не реализована	Базовый подкласс ОС. Реализован наиболее широкий спектр механизмов синхронизации и способов восстановления ВП. Некоторые способы управления ВП обеспечивают эффективное использование ресурсов МВС, но не позволяют организовать резервирование программ на уровне, требуемом для работы средств обеспечения отказоустойчивости
Реализация с использованием исполнительной системы параллельных ЯВУ	ОС, в которых возможности реализации синхронного режима ограничены вследствие как централизованного управления межмашинным обменом, так и специфики языковой реализации функции синхронизации. Отказоустойчивость МВС не реализована	ОС, в которых возможности реализации синхронного режима ограничены вследствие специфики языковой реализации функции синхронизации. Реализован один способ восстановления ВП

4.1. *Программные объекты, реализующие выполнение функций основных элементов ОС.* Для любых МВС со случайным распределением ресурсов характерно разделение системного и прикладного ПО на резидентную и перемещаемую части. Резидентная часть постоянно находится в памяти ЦВМ, перемещаемая в определенные моменты времени может перераспределяться (мигрировать) между ЦВМ. Возможность перемещения программ в ходе ВП обеспечивает равномерное распределение вычислительной нагрузки между ЦВМ и может быть использована для повышения производительности и отказоустойчивости МВС. Функции основных элементов ОС распределены между системными программами, содержащимися в резидентной и перемещаемой части ПО. Резидентная часть ПО содержит средства встроенного контроля, диспетчер и средства межмашинного обмена. Их функции такие же, как и одноименных элементов ОС МВС с детерминированным распределением ресурсов. Перемещаемая часть ПО содержит прикладные и системные задачи, сгруппированные по виртуальным машинам (ВМ). Во всех известных нам отказоустойчивых МВС резервирование программ, содержащихся в перемещаемой части ПО, и управление их перемещением реализовано на уровне ВМ. Сложная структура ВМ характерна только для базового подкласса ОС. В остальных подклассах ВМ содержит только одну системную или прикладную задачу и порты обмена. В нерезервированных МВС допустимо размещение системных программ ВМ в памяти разных ЦВМ [85]. В отказоустойчивых МВС все системные программы ВМ должны быть размещены в памяти одной ЦВМ. Копии прикладных (системных) задач должны храниться в памяти разных ЦВМ.

Определим программные объекты в резидентной и перемещаемой части ПО, выполняющие функции основных элементов ОС. Функции супервизора распределены между портами обмена, программами управления задачами в ВМ и общим для каждой ЦВМ диспетчером. *Под портом понимается коммуникационный посредник в обмене данными между задачами, используемый независимо от того, размещены эти задачи в одной или разных ВМ* [86]. Порты могут создаваться либо статическим образом (одновременно с генерацией ВМ, за которыми они постоянно закреплены), либо динамическим, в ходе выполнения задач. Динамические порты по мере изменения потребности в их использовании могут также передаваться от одной ВМ к другой или уничтожаться. Кроме операций создания/уничтожения и передачи портов, в ОС определены также операции коммутации (связывания между собой) портов ВМ [85] и обработки событий прихода сообщений в порты, по которым ожидающие их задачи должны переходить в состояние готовности к выполнению. Диспетчер выбирает (по приоритетам) из общей очереди задач, находящихся в состоянии готовности, ту, которую переводит в активное состояние при предоставлении ей ресурсов, а по окончании выполнения возвращает ее в пассивное состояние. Во всех известных отказоустойчивых МВС со случайным распределением ресурсов функции диспетчера реализованы независимо от остальных элементов ОС, с использованием индивидуальной для каждой ЦВМ службы системного времени [87, 88].

Синхронный режим межмашинного обмена реализован с помощью портов и системных задач, содержащихся в перемещаемой части ПО. Порты (системные задачи) во взаимодействии со средствами межмашинного обмена реализуют распределенный (централизованный) вариант синхронного режима. Механизм портов является многофункциональным, т.е. одновременно участвует в выполнении функций супервизора и реализует синхронный режим обмена. Разделить операции, выполняемые портами, между супервизором и ядром (механизмом синхронизации) не представляется возможным, потому что функции основных элементов ОС выполняются на более высоком уровне, с помощью взаимодействия нескольких программных объектов.

Функции гипервизора выполняются частью системных задач, содержащихся в ВМ. С их помощью во взаимодействии с портами и средствами межмашинного обмена выполняется обработка КТ. Функции основных элементов ОС МВС со случайным распределением ресурсов приведены в табл. 18. Информация в ней соответствует способу восстановления ВП в динамической системе из нескольких КРЗ. Именно этот способ был реализован во всех подклассах, в которых ОС позволяет обеспечить отказоустойчивость МВС.

4.2. *Способы реализации основных элементов ОС в базовом подклассе.* Для базового подкласса характерно многообразие механизмов синхронизации межмашинного обмена и способов восстановления ВП. Проведем анализ на примере трех систем: *ПАРУС* [85, 89], *CHORUS* [86, 90 – 92] и *CONIC* [93]. Они обладают, по-видимому, принципиально разными возможностями реализации отказоустойчивости. Авторы разработки ОС *ПАРУС* предлагают ее как универсальное средство, настраиваемое на работу в любой распределенной МВС РВ, в том числе и в отказоустойчивой. Для ОС, создаваемой на основе системы *ПАРУС*, предлагается набор языковых средств, с помощью которых могут быть разработаны программные механизмы всех основных элементов. Система *CHORUS* также трактуется как настраиваемая на конкретную область применения МВС. В ней функции основных элементов ОС реализованы несколькими способами, однако в известных публикациях не приведены завершенные примеры реализации отказоустойчивых систем. Наконец, ОС *CONIC* изначально разрабатывалась для работы только в отказоустойчивых МВС. В [94] приведены варианты реализации основных элементов ОС динамической МВС из нескольких КРЗ.

4.2.1. *Синхронный режим межмашинного обмена.* Предоставление ресурсов прикладным задачам в каждой ЦВМ осуществляется независимо. Поэтому единствен-

Таблица 18

Функции основных элементов ОС,
реализующие управление и восстановление ВП

Элемент ОС	Функции элемента ОС		
	Управление ВП	Восстановление ВП	
	Стационарный и нестационарный процессы	Стационарный процесс	Нестационарный процесс
Супервизор	Диспетчеризация в ЦВМ прикладных, системных задач и всех операций обмена данными с другими ЦВМ и с абонентами по событиям приема в порты обмена сообщений, адресованных этим задачам. Динамическое создание/уничтожение портов обмена и их перераспределение между ВМ (только в стационарном процессе, осуществимо в ОС с системной реализацией синхронного режима межмашинного обмена)	–	–
Ядро	Реализация синхронного режима межмашинного обмена	Контроль допустимого времени ожидания приема сообщений-откликов от всех ЦВМ, участвующих в обмене данными, выполняемом в синхронном режиме	Обнаружение неисправности по признаку аномального завершения попытки синхронизации
Гипервизор	–	В каждой КТ выполняются тестовая проверка результата решения активной копии прикладной задачи и контроль признаков успешного завершения межмашинного обмена	Обнаружение неисправности по признаку безуспешного завершения межмашинного обмена; реализация замещения отказавшей ВМ; создание новой копии резервной ВМ; перераспределение копий ВМ между исправными ЦВМ

ный способ синхронизировать моменты начала выполнения задач в нескольких ЦВМ состоит в выполнении обмена сообщениями между задачами с уведомлением об успешном/безуспешном его завершении. При реализации синхронного режима межмашинного обмена в отказоустойчивых МВС на низком уровне могут быть использованы программные объекты, которые в нерезервированных системах применяются для выполнения любых операций обмена: порты, средства передачи сообщений, системные задачи – серверы, уведомляющие передающую и принимающую данные ВМ об успешной/безуспешной передаче сообщения. С помощью этих объектов можно реализовать как синхронный, так и асинхронный режимы обмена между ВМ.

В любой ОС базового подкласса синхронный режим обмена может быть реализован: с помощью процедуры удаленного обслуживания запросов (*remote call procedure*) (ОС *CHORUS*); с помощью механизма надежных портов (ОС *CONIC*). Процедура удаленного обслуживания запросов реализована в ОС *CHORUS* следующим образом. Задача обращается к системному серверу с запросом обмена с задачей, размещенной в другой ВМ, и получает ответную информацию о его успешном/безуспешном завершении. В любых локальных сетях, в том числе и при взаимодействии между нерезервированными ЦВМ, после проявления неисправностей в межмашинном обмене осуществляется повторная передача сообщений. Для этого в каждую операцию обмена введены КТ с уведомлением всех ЦВМ, участвующих в ее выполнении, об успешном/безуспешном завершении обмена. В ОС *CONIC* этот механизм был усовершенствован и получил наименование надежных (*reliable*) портов (*R*-портов) [95], предназначенных для восстановления ВП в операциях обмена между КРЗ при любых одиночных отказах ЦВМ (шин). Перед началом межкомплексного обмена выполняется внутрикомплексный обмен между ВМ-источниками, с организацией точки хранения. Межкомплексная передача сообщения выполняется от активной ВМ комплекса-источника во все ВМ комплекса-приемника. Каждая ВМ комплекса-приемника передает сообщение-отклик во все ВМ комплекса-источника. В случае отказа основной ВМ в любом комплексе резервная принимает на себя ее функции и осуществляет восстановление ВП путем повторной передачи сообщения, не подтвержденного откликом. Сообщениям присвоены уникальные номера, что позволяет обнаружить дублирующие сообщения и предотвратить попытку их повторной обработки в исправных ЦВМ.

4.2.2. *Восстановление ВП.* В базовом подклассе ОС был реализован только один способ восстановления, применявшийся в системах, состоявших из нескольких КРЗ. Остальные способы представлены в виде концепций реализации некоторых функций гипервизора, но не содержат целостного описания всех шагов обработки КТ [96]. На примере ОС *CHORUS* рассмотрим практически реализованный способ восстановления ВП ("дублированные ВМ" – "*coupled actors*" [91]). Затем приведем предлагаемый в той же ОС механизм миграции системных задач гипервизора ("активные сообщения" – "*activity messages*" [90, 92]), который может быть использован при разработке перспективных способов восстановления ВП в КПР.

Восстановление ВП в системе из нескольких КРЗ. В ОС *CHORUS* две ВМ, содержащие копии прикладной задачи, размещенные в памяти разных ЦВМ и взаимодействующие по принципу "ведущий – ведомый", образуют КРЗ. Интервал времени непрерывного выполнения одной задачи называется шагом обработки (*processing step*). Передача сообщений между ВМ выполнима только после окончания шага обработки. Обе копии являются активными, но основная ВМ опережает резервную на один шаг обработки. После окончания очередного шага выполняется передача сообщения от основной ВМ к резервной с исходными данными для выполнения того же самого шага. Любая неисправность интерпретируется как проявление отказа ЦВМ, требующего ее замещения при восстановлении ВП. Обработка КТ состоит из следующих шагов: обнаружение неисправности и ее локализация; замещение основной (резервной) копии ВМ после отказа ЦВМ; создание новой резервной копии ВМ; перемеще-

ние ВМ между исправными ЦВМ для выравнивания их вычислительной нагрузки. Первые три применимы в статических и в динамических МВС, последний – только в динамических. Каждый шаг реализован во всех известных распределенных ОС. Рассмотрим способы их реализации.

Обнаружение и локализация неисправности реализованы двумя способами [89]: по превышению допустимого времени ожидания приема сообщения-отклика с признаком успешного/безуспешного завершения обмена; по частным признакам проявления неисправности, с помощью системных задач, выполняющих два типа функций. Задачи первого типа выполняют диагностику ЦВМ, обнаруживают частные признаки проявления неисправностей и генерируют сообщения; задачи второго типа в ответ на эти сообщения инициируют восстановление ВП. Оба способа позволяют с помощью диагностических программ обнаружить и локализовать неисправность лишь для части возможных случаев ее проявления, поэтому обладают более низкой эффективностью, чем механизмы, используемые в КПР (раздел 3.2.3).

Резервная ВМ замещает основную с помощью открытия портов обмена с остальными ВМ, выполняющими основные копии других задач. Имена вновь открытых портов совпадают с использованными отказавшей ВМ. Подобным же образом осуществляется замещение резервной ВМ. Создание резервной копии ВМ осуществляется либо статическим, либо динамическим способом. Статический способ используется в отказоустойчивых МВС, работающих в системах управления. В каждую ЦВМ заранее загружаются ВМ, которые могут в ней выполняться на всех допустимых (заранее заданных) уровнях деградации МВС. Каждый уровень деградации определяется сочетанием числа исправных и отказавших ЦВМ. На каждом уровне деградации любая исправная ЦВМ выполняет лишь часть хранящихся в ее памяти ВМ (остальные блокированы). Создание новой резервной копии выполняется с помощью деблокирования ВМ, которая открывает порты обмена для приема сообщений от основной ВМ. Динамический способ создания резервной копии ВМ используется только в экспериментальных системах. При этом все ВМ, хранящиеся в памяти исправных ЦВМ, являются активными. При восстановлении ВП осуществляется загрузка из высоконадежной внешней памяти в исправные ЦВМ тех ВМ, которые хранились в памяти отказавшей.

Механизм перемещения ВМ между исправными ЦВМ реализован в ОС *ПАРУС* следующим образом [89]. В любой момент времени каждая исправная ЦВМ имеет информацию о том, какие ВМ она должна будет принять на себя после отказа любой другой ЦВМ, находящейся в данный момент в исправном состоянии. Для получения такой информации каждая ЦВМ пишет "завещание" остальным исправным, в котором она определяет списки ВМ, которые должны выполняться в других ЦВМ после ее отказа. Реакция МВС на первый отказ рассчитывается статически, до начала работы системы, на последующие – динамически, по мере деградации.

Таким образом, обнаружение и локализация неисправности выполняются теми же средствами, как и в статическом КРЗ в МВС с детерминированным распределением ресурсов (раздел 3.3). Последующие шаги обработки КТ характерны для восстановления ВП в динамических системах из нескольких КРЗ и реализованы только в МВС со случайным распределением ресурсов.

Механизм миграции программ гипервизора. Механизм внутрикомплексной передачи сообщений, содержащих программы гипервизора, по замыслу разработчиков ОС *CHORUS* должен быть основой для реализации различных способов восстановления ВП в МВС, состоящей из одного или нескольких КПР [90, 92]. Реализован обмен двумя видами сообщений: обычными, содержащими адресную часть и данные, передаваемые между прикладными задачами; особыми, так называемыми "активными сообщениями", содержащими текст программы, поочередно выполняемой разными ВМ. Именно "активные сообщения" позволяют реализовать миграцию задач гипервизора. "Активное сообщение" состоит из трех частей: области хранения программы, которая представляет собой последовательность шагов обработки, по-

следовательно выполняемых разными ВМ; контекстной части, содержащей значения параметров точки входа в программу, от которой ее выполнение должно быть продолжено в ВМ, принявшей сообщение; области хранения данных, обрабатываемых в программе.

Приняв "активное сообщение", ВМ должна выполнить свой шаг обработки, определить следующий шаг, изменить содержание контекстной части и области хранения данных, после чего передать это сообщение следующей ВМ. Кроме того, ВМ может создавать версии "активного сообщения", передаваемые нескольким ВМ, или объединять несколько версий в одно сообщение. Версии обладают одинаковой программной частью и разным содержанием остальных двух частей сообщения. Задание на создание/объединение версий должно содержаться в программной части "активного сообщения". Программная часть состоит из операторов, предназначенных: для указания, в какой порт требуется передать сообщение для выполнения следующего шага обработки (*FORWARD_M*); для выполнения вложенных задач (типа подпрограмм) (*CALL*); для создания нескольких версий "активного сообщения" (*SPLIT*); для объединения версий в одно сообщение (*JOIN*). При выполнении оператора *SPLIT* исходное сообщение задерживается в ВМ и записывается в память. Вместо него запускается требуемое количество версий этого сообщения (заданное в значениях параметров оператора *SPLIT*). При выполнении оператора *JOIN* происходит слияние версий с исходным сообщением, запись в его область хранения данных информации, принятой от каждой версии, после чего возобновляется выполнение программы исходного сообщения. Допускается возможность завершения выполнения некоторых версий без прохождения через оператор *JOIN*.

Авторы разработки механизма "активных сообщений" предлагают несколько вариантов его использования в отказоустойчивых МВС [90, 92]. Например, можно реализовать выбор согласованного значения результата выполнения активных копий прикладной задачи путем последовательного обхода "активным сообщением" всех ВМ в КПР. Предлагается также организовать выполнение различных версий прикладной задачи, записанных в программной части "активного сообщения" (активизация версий с помощью оператора *SPLIT*, а проверка результатов вычислений – в исходном сообщении после выполнения оператора *JOIN*). Механизм "активных сообщений" реализует функции как внутрикомплексной синхронизации в КПР, так и гипервизора. В последующем будем интерпретировать их раздельно, при сравнительном анализе с механизмами синхронизации и со способами восстановления ВП.

Способы восстановления ВП, реализованные с помощью механизмов надежных портов и "активных сообщений", описываются разными концептуальными моделями, естественно, отличающимися от любых, применимых для МВС с детерминированным распределением ресурсов.

4.2.3. *Сравнительный анализ характера использования механизмов ОС, управляющих параллельными процессами в МВС.* Наивно полагать, что одни и те же механизмы ОС, реализующие управление параллельными процессами в МВС, всегда могут быть применимы для повышения и производительности, и отказоустойчивости. На самом деле это не так, причем именно многообразие механизмов управления параллельными процессами на различных уровнях ВП, реализованных в базовом подклассе ОС, позволяет доказать иное утверждение. Существуют способы управления ВП, позволяющие эффективно использовать ресурсы МВС для повышения производительности, но неприемлемые для реализации восстановления. Приведем подтверждающие примеры. Для этого рассмотрим реализацию в ОС статического/динамического создания программных объектов (ВМ и их портов обмена), разбиения составных объектов на части, хранимые в памяти нескольких ЦВМ, перемещения объектов в ходе ВП, а также синхронного/асинхронного режимов передачи данных.

Динамическое создание/уничтожение программных объектов, безусловно, снижает затраты памяти и, может быть, процессорного времени на решение прикладн-

ных задач по сравнению со статической их реализацией. Однако, как правило, динамические свойства реализованы при скрытом от любых системных задач (не связанных с механизмом перемещения программных объектов) характере их распределения между ЦВМ, что является неприемлемым для реализации отказоустойчивости. В отказоустойчивых системах требуется, чтобы копии любого программного объекта хранились в памяти разных ЦВМ, причем характер их распределения должен быть открытым для гипервизора. Выполнение же этого требования может быть достигнуто за счет существенных дополнительных затрат ресурсов на оповещение гипервизора о каждом перемещении объектов, которые нивелируют эффект снижения затрат.

Разбиение составных программных объектов (например, ВМ) на части, хранимые в памяти нескольких ЦВМ, потенциально может снизить затраты за счет более равномерного распределения нагрузки в МВС, однако приведет к значительному усложнению реализации отказоустойчивости. Потребуется создавать копии объектов и разработать средства обработки КТ на том же уровне, на котором осуществлено разбиение ВМ. Это потребует значительных затрат вследствие того, что невозможно использовать готовые системные программы, применяемые в комплексах ЦВМ. Поэтому при создании отказоустойчивой МВС более предпочтительно не разбивать ВМ на части, а размещать их целиком в памяти ЦВМ.

Возможность реализации перемещения программных объектов ошибочно интерпретируется некоторыми авторами разработки ОС как автоматически обеспечивающая отказоустойчивость МВС. На самом деле, возможность перемещения объектов может быть непосредственно использована только для повышения производительности МВС (за счет равномерного распределения нагрузки между ЦВМ) и более эффективной адаптации программ супервизора к условиям внешней среды, чем в системах с детерминированным распределением ресурсов. К тому же, в отказоустойчивых МВС ограничена возможность равномерного распределения нагрузки вследствие необходимости хранить копии задачи в памяти нескольких ЦВМ. Для реализации отказоустойчивости МВС явно недостаточно осуществить только возможность перемещения программных объектов между ЦВМ. Требуется эффективно реализовать все шаги обработки КТ, тогда как использование только диагностических программ для обнаружения и локализации неисправности существенно снижает отказоустойчивость по сравнению с системами, использующими для выполнения этих функций механизмы гипервизора, применяемые в КПР.

Наконец, выбор режима передачи данных между задачами, решаемыми в разных ЦВМ, существенно различается в зависимости от цели. Для повышения производительности предпочтительным является асинхронный режим, поскольку в каждой операции обмена не затрачивается время на ожидание информации, подтверждающей успешную передачу сообщения. В любой отказоустойчивой МВС, напротив, получение такой информации является необходимой частью формирования исходных данных для обработки КТ, поэтому все операции обмена должны выполняться только в синхронном режиме.

Таким образом, реализация отказоустойчивости МВС требует не только дополнительных затрат, но также ограничивает область допустимых вариантов распараллеливания программных объектов и их динамического перемещения между комплексами ЦВМ.

4.3. *Анализ реализации основных элементов ОС МВС с централизованным управлением межмашинным обменом.* Несмотря на то, что централизованный вариант реализован проще, чем распределенный, число ОС, в которых этот вариант используется, сравнительно невелико, причем только в некоторых из них и лишь потенциально заложена возможность реализации отказоустойчивости. Отсутствие ОС с централизованным управлением межмашинным обменом, практически обеспечивающих отказоустойчивость, вызвано следующими причинами: область применения прикладных задач, требующих только централизованного управления, чрезвычайно

узка; резервирование системных задач, осуществляющих управление, во всех известных ОС практически не реализовано. Централизованный характер потребовал использования неоднородной структуры МВС и ее ОС. Одна из ЦВМ выполняла функции ведущей, остальные – ведомых, причем в известных системах ведущая обладала более высокими характеристиками производительности. В структуре ОС особую группу образовали системные задачи, выполнявшиеся в ведущей ЦВМ. Примерами централизованных ОС с системной реализацией синхронного режима межмашинного обмена являются системы *MIKE* [87] и *SYLVAN* [97]. Примером централизованной ОС с языковой реализацией синхронизации межмашинного обмена является экспериментальная ОС, использующая исполнительную систему ЯВУ *Modula-2* [88].

Из всех функций основных элементов ОС отказоустойчивых МВС была рассмотрена только одна – реализация синхронного режима обмена между прикладными задачами, выполняемыми в ведомых ЦВМ. Во всех ОС предлагался единственный вариант ее выполнения – процедура удаленного обслуживания запросов. При этом ОС, использовавшие собственные системные программы для реализации синхронизации, отличались от использующих исполнительную систему параллельных ЯВУ способностью динамического создания/уничтожения процессов и портов. В языковых ОС использовалось только статическое их формирование.

Рассмотрим способ реализации процедуры удаленного обслуживания запросов в ОС *MIKE* [87]. Все задачи (системные и прикладные) по характеру выполнения в МВС были разделены на "временные" ("*transient*") и "циклические" ("*cyclic*"). "Временные" задачи создавались в ходе ВП и по окончании своей работы удалялись из системы. "Циклические" задачи постоянно присутствовали в системе. Естественно, они находились на более высоком уровне иерархии управления ВП, чем "временные". Все "циклические" задачи были системными, тогда как "временные" могли быть либо системными, либо прикладными. "Временные" задачи взаимодействовали между собой посредством запросов, передававшихся через "циклические" задачи. "Циклическая" задача по запросу от одной "временной" задачи могла осуществить запуск выполнения другой или организовать обмен данными между ними. "Циклические" задачи были размещены как в ведущей, так и в ведомых ЦВМ, "временные" же – только в ведомых. Любой запрос, передававшийся между "временными" задачами, размещенными в разных ведомых ЦВМ, проходил контроль "циклическими" задачами в ведущей. Резервирование было реализовано только для "временных" задач. После отказа ведомой ЦВМ "циклическая" задача в ведущей активизировала пассивные копии "временных" задач, решавшихся в отказавшей, на исправных ведомых ЦВМ, и передавала им исходные данные. В то же время ОС *MIKE* не обеспечивала восстановление после отказа ведущей ЦВМ, поскольку решавшиеся в ней "циклические" задачи не были резервированы.

4.4. *Анализ основных элементов ОС в системах с языковой реализацией синхронизации межмашинного обмена.* Все отказоустойчивые МВС, ОС которых была реализована с использованием исполнительной системы параллельных ЯВУ, имели распределенную структуру управления межмашинным обменом. Языковые ограничения способов реализации синхронного режима межмашинного обмена были связаны со спецификой конкретных ЯВУ. Поэтому выводы из анализа механизмов синхронизации могут быть применимы также к системам с централизованным управлением. Выводы же из анализа восстановления ВП относятся только к системам с распределенным управлением обменом. Механизмы синхронизации, реализованные в любом параллельном ЯВУ, организуют взаимодействие между задачами (процессами). Поэтому резервирование программных объектов реализовано на уровне задач.

4.4.1. *Анализ реализации синхронного режима межмашинного обмена.* Использование языковых механизмов синхронизации параллельных процессов для реализации синхронного режима межмашинного обмена создает ограничения в обеспечении отказоустойчивости МВС, имеющие общий и частный характер. Общими огра-

ничениями являются: отсутствие средств контроля физического размещения задач между ЦВМ (для любого параллельного ЯВУ они могут содержаться только в ОС); возможность использования только статической организации портов обмена между прикладными задачами (что ограничивает возможности снижения затрат памяти, но не влияет на реализацию отказоустойчивости); отсутствие широковещательной передачи сообщений от одной к нескольким задачам (что вынуждает формировать систему только из КРЗ).

Частные ограничения специфичны для конкретных ЯВУ и характеризуют объем случаев проявления неисправности ЦВМ в виде аномального завершения попытки межмашинной синхронизации, обнаруживаемых с помощью языковых механизмов. Приведем примеры анализа частных ограничений для двух ЯВУ: *Ada* и *Occam*. Язык *Ada* разрабатывался независимо от аппаратуры МВС. Имеющийся в нем механизм синхронизации обладает ограниченными возможностями обнаружения аномального завершения попытки взаимодействия процессов. В ЯВУ *Ada* для обнаружения аномального завершения синхронизации параллельных процессов определена исключительная ситуация *TASKING_ERROR*. Пусть задача A вызывает выполнение задачи B. Задача A должна продолжить свое выполнение только после окончания задачи B. Исключительная ситуация *TASKING_ERROR* в обмене между ними может быть вызвана следующими причинами [98]: невозможность запуска задачи B; к моменту запроса задача B уже была закончена или находилась в состоянии аварийного завершения. Если во время взаимодействия возникнет аварийное завершение в задаче B, то оно распространится на задачу A. Если же аварийное завершение возникнет в задаче A, то это не повлияет на задачу B. Таким образом, асимметричность возможностей обнаружения аномального завершения взаимодействия между задачами в ЯВУ *Ada* позволяет обнаружить с помощью его механизмов лишь часть случаев проявления неисправностей ЦВМ при выполнении попытки межмашинной синхронизации. Язык *Occam* был разработан для использования в сети транспьютеров [99 – 107]. В нем параллельные процессы взаимодействуют с помощью посредников, называемых "каналами" (аналог портов обмена). Как запрашивающая, так и обслуживающая запрос задачи обращаются к одному и тому же "каналу" и одинаковым образом распознают ситуацию аварийного завершения обмена. Ошибки при передаче данных между транспьютерами обнаруживаются с помощью стандартных процедур *InputOrFail* и *OutputOrFail*, результатом выполнения которых является значение булевой переменной, означающей успешное/безуспешное завершение соответственно приема или передачи сообщения [108]. Повторная передача сообщений в случае безуспешного завершения осуществляется с помощью стандартной процедуры *Reinitialise*. Таким образом, именно благодаря тому, что язык *Occam* изначально был разработан для применения в МВС, использование механизма синхронизации параллельных процессов не создает частных ограничений при реализации синхронного режима межмашинного обмена.

4.4.2. *Анализ способов восстановления ВП.* Известны только два примера практической реализации отказоустойчивости МВС с помощью распределенной ОС, использующей языковые механизмы синхронизации параллельных процессов для осуществления синхронного режима межмашинного обмена. Одна из систем реализована с использованием ЯВУ *Ada* [94], вторая – ЯВУ *Occam* [95]. В обеих ОС реализован один и тот же способ восстановления в МВС из нескольких КРЗ (раздел 4.2.2). Отличия от ОС, использующих собственные системные программы для реализации синхронного режима межмашинного обмена, проявились только во вспомогательных программных средствах, поддерживающих работу гипервизора. Эти средства предназначены для преодоления языковых ограничений, рассмотренных в предыдущем разделе.

В ОС, реализованной на основе исполнительной системы ЯВУ *Ada*, функции гипервизора выполнялись с помощью системных задач, осуществлявших обнаружение

Таблица 19

Сравнительный анализ степени достижимой отказоустойчивости в МВС со случайным распределением ресурсов

Требования к ОС, определяющие степень достижимой отказоустойчивости МВС	Варианты реализации механизма синхронизации, определяющие подклассы ОС			
	Централизованная структура управления межмашинным обменом		Распределенная структура управления межмашинным обменом	
	Системная реализация синхронизации	Языковая реализация синхронизации	Системная реализация синхронизации	Языковая реализация синхронизации
Возможность реализовать резервирование любых системных и прикладных задач	–	–	+	+
Возможность реализовать механизм синхронизации, позволяющий осуществить взаимодействие между тремя или более высоким числом активных копий прикладных задач	–	–	+	–

неисправностей, перевод пассивных копий прикладных задач в активное состояние, реконфигурацию внутри- и межкомплексных связей после отказов ЦВМ. Вспомогательные системные задачи, поддерживавшие работу гипервизора, выполняли следующие функции. Они реализовали открытое для него распределение прикладных задач между ЦВМ, а также осуществляли обработку сообщений, передававшихся между задачами гипервизора. Так, например, запуск восстановления ВП осуществлялся с помощью сообщений, передаваемых от диагностических программ (*fake messages*).

Для большинства ОС, использующих исполнительную систему ЯВУ *Occam*, характерны аппаратурные ограничения при реализации отказоустойчивости. Они проявляются косвенным образом: чем больше емкость ОЗУ ЦВМ, из которых формируются комплексы, тем более развитым образом осуществляется в них восстановление ВП. Эта зависимость характерна только для ЦВМ с малой емкостью ОЗУ, в частности для серии транспьютеров первого выпуска. Эта серия *T-414* имела очень малую емкость ОЗУ – только 4 Кбайт [106]. Отказоустойчивость в системе из таких транспьютеров могла быть реализована только простейшими средствами. В отказоустойчивой бортовой МВС, рассмотренной в разделе 3.7, были использованы транспьютеры серии *T-222*, емкость ОЗУ каждого из них составляла уже 62 Кбайт. Поэтому в этой системе была допустима реализация способов восстановления, применимых в КПР в системах с детерминированным распределением ресурсов. Наконец, в системе из транспьютеров перспективной серии *T-9000* предполагается использовать ОС *CHORUS* [86]. В системе, представленной в [95], аппаратурные

ограничения проявились в наименьшей степени. В ней КРЗ были представлены в явном виде (как в любых статических отказоустойчивых МВС). Восстановление ВП после неисправностей, проявлявшихся в межмашинном обмене, было реализовано способом, который применялся при реализации надежных портов в ОС *CONIC* (раздел 4.2.1).

4.5. *Сравнительный анализ степени достижимой отказоустойчивости МВС со случайным распределением ресурсов.* Степень достижимой отказоустойчивости будем оценивать по признаку достаточности реализованных вариантов резервирования системных и прикладных задач для восстановления ВП либо только одним, либо несколькими способами. Результаты сравнительного анализа приведены в табл. 19.

Дадим необходимые пояснения. Для обеспечения отказоустойчивости необходимо, чтобы любые (как прикладные, так и системные) задачи, выполняемые в МВС, были резервированы. Возможность резервирования прикладных задач реализована в любом подклассе ОС МВС со случайным распределением ресурсов. Возможность же резервирования системных задач, реализующих синхронный режим межмашинного обмена, осуществлена только в ОС с распределенной структурой управления обменом. Таким образом, минимально необходимые требования резервирования любых задач, позволяющие реализовать хотя бы один способ восстановления ВП (сформировать систему из нескольких КРЗ), удовлетворены только в МВС с распределенной структурой управления обменом.

Возможность реализации в ОС нескольких способов восстановления ВП осуществима в том случае, если реализованный в ней механизм синхронизации позволяет осуществлять обмен данными между тремя или более ЦВМ. Только в этом случае возможно реализовать хотя бы часть спектра способов восстановления, применяемых в статическом синхронном КПР с детерминированным распределением ресурсов, так как большинство из них требуют кратность резервирования ЦВМ не менее трех. Из всех исследованных механизмов синхронизации данному требованию удовлетворяет механизм "активных сообщений". Принципиальная возможность разработки других подобных механизмов существует только в базовом подклассе ОС.

Таким образом, наиболее предпочтительными для использования в отказоустойчивых МВС являются распределенные ОС с системной реализацией синхронизации; наименее предпочтительными – централизованные ОС. Промежуточное положение занимают распределенные ОС с языковой реализацией синхронизации.

5. Сравнительный анализ поддержки отказоустойчивости элементами ОС, управляющими ВП

Управляющие отказоустойчивые МВС с детерминированным и случайным распределением ресурсов существенно различаются по способам управления и восстановления ВП. Наиболее широкий спектр способов восстановления реализован в МВС с детерминированным распределением ресурсов. В системах со случайным распределением ресурсов восстановление реализовано практически только одним способом. Это объяснимо различной степенью поддержки отказоустойчивости элементами ОС, осуществляющими управление ВП. Проведем анализ поддержки отказоустойчивости всеми механизмами супервизора и ядра, приведенными в обзоре. Для каждого элемента ОС он состоит из формирования требований к способам реализации функций, необходимых для поддержки отказоустойчивости; анализа условий, необходимых для удовлетворения этих требований механизмами супервизора (ядра); формулирования выводов о предпочтительном характере распределения ресурсов, необходимом для того, чтобы поддержка отказоустойчивости была эффективной. В заключение сделаем выводы о предпочтительных характеристиках управления ВП, в целом способствующих реализации отказоустойчивости.

5.1. *Анализ поддержки отказоустойчивости, реализуемой супервизором.* В любой нерезервированной МВС функции супервизора выполняются, как правило, в

Таблица 20

Потенциальная поддержка отказоустойчивости супервизором в МВС с детерминированным и случайным распределением ресурсов

Требования к реализации функций супервизора, необходимые для поддержки отказоустойчивости МВС	Реализуемость требований поддержки отказоустойчивости МВС супервизором	
	В МВС с детерминированным распределением ресурсов	В МВС со случайным распределением ресурсов
Открытый для локального гипервизора характер распределения копий прикладных задач между ЦВМ в динамической системе из нескольких комплексов	Реализуется благодаря резидентному хранению в памяти каждой ЦВМ системных и прикладных программ, априорно составленному расписанию их выполнения и единой службе системного времени во всей МВС	Реализуется с помощью передачи сообщений от каждой ЦВМ, содержащих перечень выполняемых ею задач. Сообщения должны быть переданы при каждом перераспределении активных копий задач между ЦВМ
Одновременное для всех ЦВМ комплекса (системы) предоставление ресурсов прикладным задачам или любым операциям обмена, которые должны быть выполнены в синхронном режиме	Реализуется с помощью единой службы системного времени и временного механизма диспетчеризации синхронизируемых операций обмена	В механизмах диспетчеризации поддержка синхронного режима отсутствует. Синхронный режим реализован с помощью механизма портов обмена данными
Детерминированный характер предоставления ресурсов прикладным задачам и операциям внутрикомплексного обмена, необходимый для реализации управления ресурсами комплекса на интервале восстановления ВП	Осуществляется таким же способом, как при реализации синхронного режима внутрикомплексного обмена	Эта функция ядра в известных нам МВС со случайным распределением ресурсов не была, и, возможно, не могла быть реализована

каждой ЦВМ независимо от остальных. В отказоустойчивых МВС, по крайней мере во время восстановления ВП, функции локального (глобального) супервизора должны выполняться согласованно всеми ЦВМ комплекса (системы). Согласованная работа супервизоров требует: открытый для локального гипервизора характер распределения копий прикладных задач между ЦВМ в динамической системе; одновременное для всех ЦВМ комплекса (системы) предоставление ресурсов прикладным задачам или любым операциям обмена, которые при заданном способе восстановления должны выполняться в синхронном режиме; детерминированный характер предоставления ресурсов прикладным задачам и операциям внутрикомплексного обмена. Все эти требования могут быть удовлетворены только в МВС с детерминированным распределением ресурсов с помощью единой службы системного времени и априорно составленного расписания выполнения прикладных задач (табл. 20).

В системах со случайным распределением ресурсов механизмы диспетчеризации прикладных задач не могут оказать поддержку отказоустойчивости вследствие независимой для каждой ЦВМ службы системного времени. В этом классе МВС с

Таблица 21

Возможность удовлетворения требований поддержки отказоустойчивости механизмами межмашинной синхронизации, реализованными в МВС с детерминированным и случайным распределением ресурсов

Механизмы синхронизации и область их возможной реализации в МВС		Требования к способу реализации синхронного режима межмашинного обмена, поддерживающие отказоустойчивость МВС	
Наименование механизма	Характер распределения ресурсов в МВС, в которых реализован механизм	Резервирование аппаратурно-программных средств, выполняющих синхронизацию	Возможность использования механизма для внутрикомплексной синхронизации в КПР с кратностью резервирования ЦВМ не менее трех
1. Синхронизация по сигналам таймера при наличии единой службы системного времени	Детерминированный	+	+
2. Механизм упорядочения последовательности обработки в КПР асинхронных сигналов внешних прерываний	Детерминированный	+	+
3. Механизм надежных портов	Детерминированный или случайный с распределенной структурой управления обменом	+	–
4. Механизм удаленного обслуживания запросов	Случайный с централизованной структурой управления обменом	–	–
5. Механизм "активных сообщений"	Случайный с распределенной структурой управления обменом	–	+

помощью механизма синхронизации, реализуемого независимо от супервизора, могут быть удовлетворены только первые два требования. Таким образом, детерминированный характер распределения ресурсов создает естественные условия для поддержки супервизором отказоустойчивости МВС. В системах со случайным распределением ресурсов поддержка отказоустойчивости требует существенных накладных расходов в виде вспомогательных операций межмашинного обмена и потому реализуется значительно менее эффективно.

5.2. *Анализ поддержки отказоустойчивости, реализуемой ядром.* Единственная функция ядра, реализованная в МВС как с детерминированным, так и со случайным распределением ресурсов, осуществляет синхронный режим межмашинного обмена. В отказоустойчивых МВС реализация синхронного режима должна удовлетворять следующим требованиям: аппаратурно-программные средства, выполняющие синхронизацию, должны быть резервированы с кратностью, равной числу ЦВМ, одновременно участвующих в межмашинном обмене; для того чтобы дать возможность использования в КПР механизмов локального гипервизора, для реализации которых требуется кратность резервирования активных копий прикладных задач не менее трех, механизмы ядра должны реализовать синхронное взаимодействие требуемого числа ЦВМ. Всем требованиям удовлетворяет любой механизм синхронизации, реализованный в МВС с детерминированным распределением ресурсов (табл. 21).

Большинство же механизмов синхронизации, реализованных в МВС со случайным распределением ресурсов, не являются полностью резервированными и организуют взаимодействие только между двумя ЦВМ, что является недостаточным для организации ВП в КПР, за исключением дублированного комплекса. Например, в механизме "активных сообщений" операции создания/уничтожения копий сообщения являются нерезервированными. Наиболее универсальным из них является механизм надежных портов, поскольку удовлетворяет первому требованию и может быть использован для синхронизации обмена в системе, состоящей либо из КРЗ, либо из дублированных КПР при любом характере распределения ресурсов.

Невозможность удовлетворения всех требований поддержки отказоустойчивости любым механизмом синхронизации, реализованным в МВС со случайным распределением ресурсов, объяснима тем, что изначально он должен был использоваться для управления параллельными нерезервированными процессами. Закономерно, что эффективная реализация механизма синхронизации либо для управления параллельными процессами, либо для обеспечения отказоустойчивости достигается принципиально разными способами.

Был проведен анализ использования исполнительной системы параллельных ЯВУ в ОС отказоустойчивых МВС с детерминированным и случайным распределением ресурсов. Выяснилось, что специфика языковой реализации синхронизации влияет на отказоустойчивость МВС только в том подклассе, в котором с ее помощью реализован механизм надежных портов. Естественно, что в тех подклассах, в которых либо синхронизация реализована независимо от ЯВУ (раздел 3.7), либо применяемый языковый механизм не может быть резервирован (раздел 4.3), специфика исполнительной системы ЯВУ не влияет на обеспечение отказоустойчивости.

5.3. *Выводы о характере поддержки отказоустойчивости механизмами ОС, организующими ВП.* Анализ характера поддержки отказоустойчивости механизмами супервизора и ядра позволяет сделать следующие выводы.

1. Реализация отказоустойчивости предъявляет требования к способу управления ВП в МВС. Распределение копий прикладных задач между ЦВМ должно быть открытым для гипервизора. По крайней мере во время восстановления ВП характер распределения ресурсов в комплексе должен быть детерминированным, а внутрикомплексный обмен предпочтительно выполнять в синхронном режиме. Межкомплексный обмен должен выполняться только в синхронном режиме.

2. К способам реализации механизмов синхронизации либо параллельных процессов в нерезервированных МВС, либо внутрикомплексного (межкомплексного) обмена в отказоустойчивых системах предъявляются существенно различные требования. Поэтому предпринятая в ОС МВС со случайным распределением ресурсов попытка использовать механизмы, управляющие параллельными процессами, для синхронизации обмена привела к заведомо неэффективной реализации отказоустойчивости.

6. Заключение

Представленные в обзоре МВС занимают относительно узкую часть спектра известных способов управления параллельными вычислениями. Тем не менее, по тем системам, в которых практически реализована аппаратурно-программная отказоустойчивость, спектр возможных способов реализации ВП представлен полностью (естественно, за исключением информационных систем).

Зависимость способа восстановления ВП от способа управления им имеет место в любой отказоустойчивой МВС, в которой реализована стратегия локализации. Игнорирование ее приводило к ошибочному определению области возможной реализации способов восстановления ВП в управляющих и информационных системах. Установлено, что, в отличие от информационных систем, в управляющих МВС не существует единой концептуальной модели, связывающей способы управления и восстановления ВП. Число концептуальных моделей равно числу способов управления ВП, определяемых следующими характеристиками: характером распределения ресурсов между прикладными задачами (детерминированным или случайным); структурой управления межмашинным обменом (централизованной или распределенной); характером распределения прикладных задач между комплексами (статическим или динамическим); вариантом резервирования ЦВМ в комплексах (КПР или КРЗ); режимом управления ресурсами комплекса при восстановлении ВП (ПВ или ВН).

Единственным универсальным способом восстановления, используемым в МВС как с детерминированным, так и со случайным распределением ресурсов, является способ, реализованный в КРЗ. Эффективность реализации отказоустойчивости этим способом заведомо ниже, чем любым из применяемых в КПР. Поэтому любые попытки разработать ОС универсального назначения для всех управляющих МВС, настраиваемую на конкретные характеристики прикладных задач и требования по обеспечению отказоустойчивости (например, предпринимавшиеся в рамках Ньюкастловского проекта [9]), неизбежно ведут к низкой эффективности восстановления ВП.

Реализация высокой эффективности работы ОС либо по уровню производительности МВС, либо по степени ее отказоустойчивости является двойственной задачей. Улучшение одной характеристики качества системы, начиная с некоторого уровня, может быть достигнуто только за счет ухудшения второй. Высокая степень отказоустойчивости, достигнутая в динамических МВС из нескольких КПР с детерминированным распределением ресурсов, привела к высоким затратам процессорного времени на выполнение функций ОС. Высокая эффективность управления параллельными процессами в МВС со случайным распределением ресурсов привела к снижению их отказоустойчивости. Следовательно, соотношение уровней сложности алгоритмов, реализующих управление и восстановление ВП в управляющих МВС, подчиняется "правилу рычага". Чем сложнее реализованы средства управления (восстановления) ВП, тем проще должны быть реализованы средства его восстановления (управления).

В заключение авторы приносят глубокую благодарность Э. А. Трахтенгерцу за внимание, проявленное к этой работе, и конструктивную критику.

СПИСОК ЛИТЕРАТУРЫ

1. *Bena G., Frennui I.* Fault tolerance // Korp. fiz. kut. intez. Repr. 1986. № 75/M. P. 1–37.
2. *Anderson T., Lee P. A.* Fault tolerance: principles and practice. Englewood Cliffs: Prentice-Hall, 1981.
3. *Toy W. N.* Fault-tolerant computing // Advances in Computers. 1987. V. 26. P. 201–279.

4. *Malaiya Y. K., Su S. Y. H.* Fault-tolerance in multiple processor systems // Proc. 1st IEEE Int. Conf. on Circuits and Computers, 1980. P. 710–716.
5. *Мамзелев И. А., Русаков М. Ю., Часовников Е. Д., Николаенко Н. Н.* Отказоустойчивые вычислительные системы // Зарубежная радиоэлектроника. 1983. № 11. С. 3–28.
6. *Randell B.* Reliable computing systems // Lect. Notes Comp. Sci. 1978. V. 60. P. 282–391.
7. *Kim K. H., Yang S. M.* Fault tolerance mechanisms in real-time distributed operating systems: an overview // Pacific Computer Communications'85. Proc. 1st Symp. Seoul, Oct. 1985. P. 239–248.
8. *Schmid H., Larimer S., Madak T.* Channalized or nonchannalized fault-tolerant computers. A hardware complexity comparison of fault-tolerant computers for flight control systems // Proc. 7th Digital Avionics Systems Conf., 1986. P. 655–663.
9. Reliable computer systems. Collected papers of the Newcastle reliability project / Ed. by S.K. Shrivastava. Berlin; Heidelberg: Springer-Verlag, 1985.
10. *Гостилова С. В., Никитин А. И.* Глобальное состояние распределенной базы данных и глобальная контрольная точка // УСиМ. 1991. № 8. С. 68–76.
11. *Авиженис А.* Отказоустойчивость – свойство, обеспечивающее постоянную работоспособность цифровых систем // ТИИЭР. 1978. Т. 66. № 10. С. 5–25.
12. *Флинн.* Сверхбыстродействующие вычислительные системы // ТИИЭР. 1966. Т. 54. № 12. С. 311–320.
13. *Pulkkis G.* Performance aspects of the fault-tolerant multimicroprocessor μC^* // Proc. FTSD. Brno, 1979. P. 57–63.
14. *Aspelund J.* Design of an operating system to support hybrid redundant computing in a fault-tolerant multimicroprocessor system // Proc. FTSD. Brno, 1979. P. 28–34.
15. *Aspelund J.* A symmetric operating system for the hybrid redundant multimicroprocessor μC^* // Proc. 5th EUROMICRO Symp. Microprocessors and their applications, 1979. P. 147–151.
16. *Huhtanen T., Lehtinen T., Nikkila S., Pulkkis G.* μC^* – a hybrid redundant multimicroprocessor // Proc. 5th EUROMICRO Symp. Microprocessors and their applications, 1979. P. 285–293.
17. *Wensley J. H.* On-line repair of a fault-tolerant computer // Proc. American Control Conf., 1984. V. 2. P. 1107–1111.
18. *Уэнсли Дж. Х.* Высоконадежная система с тройным резервированием для управления технологическими процессами // Электроника. 1983. № 2. С. 32–39.
19. *Wensley J. H.* An operating system for a TMR fault-tolerant system // Proc. FTCS-13, 1983. P. 452–455.
20. *Yoneda T., Kawamura T., Furuya K., Tohma Y.* Fault diagnosis and system reconfiguration of fault-tolerant system with majority voting // Systems and Computers in Japan. 1985. V. 16. № 1. P. 9–17.
21. *Лорин Г., Дейтел Х. М.* Операционные системы. М.: Финансы и статистика, 1984.
22. *Майерс Г. Д.* Архитектура современных ЭВМ. Кн. 1. М.: Мир, 1985.
23. *Мартин Дж.* Вычислительные сети и распределенная обработка данных. М.: Финансы и статистика, 1986.
24. *Мамедли Э. М., Кузьмишкин С. С.* Организация вычислений в отказоустойчивых многомашинных вычислительных системах реального времени // Вопросы кибернетики. Отказоустойчивые многомашинные и многопроцессорные вычислительные системы. М.: Научный совет по комплексной проблеме "Кибернетика" АН СССР, 1989. С. 36–59.
25. *Lala J. H., Alger L. S., Gaunthier R. J., Dzwonczyk M. J.* A fault tolerant processor to meet rigorous failure requirements // Proc. 7th Digital Avionics Systems Conf., 1986. P. 555–562.
26. *Yoneda T., Suzuoka T., Tohma Y.* Interrupt handling in the loosely synchronized TMR system // Systems and computers in Japan. 1985. V. 16. № 5. P. 50–59.
27. *Pease M., Shostak R., Lamport L.* Reaching agreement in the presence of faults // J. of the ACM. 1980. V. 27. № 2. P. 228–234.

28. *Marx M. F.* Computer redundancy for aircraft flight control // Proc. Computers in Aerospace Conf., 1977. P. 329–337.
29. *Yousey W. J., Arabjan A. M., Schindler T. M., Whitmayer R. A.* AFTI/F-16 DFCS development summary – a report to industry. Redundancy management system design // Proc. NAECON-83. Dayton, 1983. P. 1220–1226.
30. *Schindler T. M., Johnston A. M., Keith G. W.* AFTI/F-16 DFCS development summary – a report to industry. Software design / implementation // Proc. NAECON-83. Dayton, 1983. P. 1227–1234.
31. *Arabjan A. M., Naumann E. A., Swihart D. E.* AFTI/F-16 digital flight control computer design // Proc. NAECON-83. Dayton, 1983. P. 1426–1432.
32. *Caulfield J. T.* Application of redundant processing to Space Shuttle // Proc. 8th IFAC Triennal World Congress. Control Science and Technology. Kyoto, 1981. V. 4. P. 2461–2465.
33. *Glazer S. D.* Fault-tolerant mini needs enhanced operating system // Computer design. 1984. V. 23. № 9. P. 189–198.
34. *McSharry M. E., McFarland M. D.* Triplex bus-connected inter-unit selected FCS configuration // Proc. NAECON-84. Dayton, 1984. V. 1. P. 645–651.
35. *Хопкинс А. Л., Смит Т. Б., Лала Дж. Х.* FTMP – высоконадежный отказоустойчивый мультипроцессор для управления самолетом // ТИИЭР. 1978. Т. 66. № 10. С. 142–165.
36. *Lala J. H.* Fault detection, isolation and reconfiguration in FTMP: methods and experimental results // Proc. 5th Digital Avionics Systems Conf., 1983. P. 21.3.1–21.3.9.
37. *Pradhan D. K., Hanquan Z., Schlumberger M. L.* Fault-tolerant multibus architecture for multiprocessors // Proc. FTCS-14, 1984. P. 400–408.
38. *Уэнсли Дж. Х., Лэмпорт Л., Голдберг Дж. и др.* SIFT: Проектирование и анализ отказоустойчивой вычислительной системы для управления полетом летательного аппарата // ТИИЭР. 1978. Т. 66. № 10. С. 166–185.
39. *Weinstock C. B., Green M. W.* Reconfiguration strategies for the SIFT fault-tolerant computer // Proc. COMPSAC-78, 1978. P. 645–650.
40. *Hitt E. F., Eldredge D.* Fault detection, isolation and recovery techniques for fault tolerant digital avionics // Proc. 5th Digital Avionics Systems Conf., 1983. P. 16.1.1–16.1.8.
41. *Генинсон Б. А., Панкова Л. А., Трахтенгерц Э. А.* Отказоустойчивые методы обеспечения взаимной информационной согласованности в распределенных вычислительных системах // АиТ. 1989. № 5. С. 3–18.
42. *Alger L. S., Lala J. H.* A real time operating system for a nuclear plant computer // Proc. Real Time Systems Symp., 1986. P. 244–248.
43. *Lala J. H.* A byzantine resilent fault tolerant computer for nuclear plant applications // Proc FTCS-16, 1986. P. 338–343.
44. *Мамедли Э. М., Самедов Р. Я., Соболев Н. А.* Метод локализации "дружественных" и "враждебных" неисправностей // АиТ. 1992. № 5. С. 126–138.
45. *Jurica K. E.* Sequencing design for BFS engagement // Proc. AIAA III Computers in Aerospace Conf., 1981. P. 150–155.
46. *Ichikawa S., Kawada Y., Mino M., Itsukaichi A.* Fault tolerant computing system, onboard computing and software for Engineering Test Satellite VI (ETS-VI) attitude control subsystem // Proc. 8th Digital Avionics Systems Conf., 1988. Pt. 1. P. 170–176.
47. *Williams R. D., Johnson B. W., Roberts T. E.* An operating system for a fault–tolerant multiprocessor controller // IEEE Micro. 1988. V. 8. № 4. P. 18–29.
48. *Julich P. M., Johnson B. W.* The fault tolerant computer system for the A129 lightweight combat helicopter // Proc. AIAA Computers in Aerospace V Conf., 1985. P. 259–266.
49. *Kirrmann H.* Fault-tolerant issues in the design of a highly available high-speed controller for HVDC transmission // Proc. FTCS-16, 1986. P. 184–189.
50. *Кравец Г. Э.* Отказоустойчивая система управления и контроля для скоростной подземной железной дороги // Электроника. 1984. № 10. С. 67–72.

51. *Yount L. T., Liebel K. A., Hill B. H.* Fault effect protection and partitioning for fly-by-wire / fly-by-light avionic systems // Proc. AIAA V Computers in Aerospace Conf., 1985. P. 275–284.
52. *Spector A., Gifford D.* The Space Shuttle primary computer system // Communications of the ACM. 1984. V. 27. N№ 9. P. 874–900.
53. *Sklaroff J. R.* Redundancy management technique for Space Shuttle computers // IBM J. on Research and Development. 1976. V. 20. N№ 1. P. 20–28.
54. *Lala J. H.* Advanced Information Processing System: fault detection and error handling // Proc. AIAA Guidance and Control Conf., 1985. P. 587–596.
55. *Lala J. H., Adams S. J.* Intercomputer communication architecture for a mixed redundancy distributed system // J. of Guidance, Control and Dynamics. 1989. V. 12. N№ 4. P. 539–547.
56. *Brock L. D., Lala J. H.* Advanced Information Processing System: status report // Proc. NAECON-86, 1986. V. 2. P. 368–375.
57. *Lala J. H.* Advanced Information Processing System // Proc. 6th Digital Avionics Systems Conf., 1984. P. 199–210.
58. *DeWolf J. B., Sodano N. M., Whittredge R. S.* Using Ada for a distributed fault-tolerant system // Proc. 6th Digital Avionics Systems Conf., 1984. P. 477–484.
59. *Schmid H.* Critical issues in the design of a reconfigurable flight control computer // Proc. 5th Digital Avionics Systems Conf., 1983. P. 16.3.1–16.3.6.
60. *Schmid H., Lam J., Naro R., Weir K.* Critical issues in the design of a reconfigurable control computer // Proc. FTCS-14, 1984. P. 36–41.
61. *Maencher H.* Synchronization tools and a restart method in the fault-tolerant distributed automation system FIPS // Informatik Fachberichte. 1984. V. 84. P. 280–291.
62. *Wensley J. H., Green M. W., Levitt K. N. et al.* The design analysis and verification of the SIFT fault tolerant system // Proc. 2nd Int. Conf. on Software Engineering, 1976. P. 458–469.
63. *Weinstock C. B.* SIFT: system design and implementation // Proc. FTCS-10, 1980. P. 75–77.
64. *Murray N. D., Hopkins A. L., Wensley J. H.* Highly reliable multiprocessors // AGARDograph N№ 224. 1977. P. 17.1–17.17.
65. *Wensley J. H.* SIFT – software implemented fault tolerance // Proc. Fall Joint Computer Conf., 1972. P. 243–253.
66. *Goldberg J.* The software-implemented fault tolerance (SIFT) approach to fault tolerant computing // Proc. Soc. Photo Opt. Instrum. Eng. 1981. V. 298. P. 289–293.
67. *Bernhard R.* The 'no-downtime' computer // IEEE Spectrum. 1980. V. 17. N№ 9. P. 33–37.
68. *Moses K.* SIFT – an ultrareliable avionic computing system // AGARD Conf. Proc. N№ 303. 1981. P. 32.1–32.10.
69. *Forman P., Moses K.* SIFT multiprocessor architecture for software implemented fault tolerance flight control and avionics computers // Proc. 3rd Digital Avionics systems Conf., 1979. P. 325–329.
70. *Goldberg J.* SIFT – a provable fault-tolerant computer for aircraft flight control // Proc. Information Processing-80 IFIP World Congress, 1980. P. 151–156.
71. *Goldberg J.* The SIFT computer and its development // Proc. 4th Digital Avionics Systems Conf., 1981. P. 285–289.
72. *Smith T. B., Hopkins A. L., Taylor W. et al.* A fault tolerant multiprocessor architecture for aircraft. US NASA contractor report N№ 3010, 1978.
73. *Elson B. M.* USAF studies new computer concept // Aviation Week and Space Technology. 1983. V. 118. N№ 19. P. 69–71.
74. *Thompson D. B., Bortner R. A.* AF multiprocessor flight control architecture developments: CRMMFCS and beyond // Proc. NAECON-86. Dayton, 1986. P. 376–382.
75. *Bortner R. A.* Continuous reconfiguration: fault tolerance without a ripple // Proc. COMPCON-83. P. 299–301.
76. *Kiekhafer R. M., Walter C. J., Finn A. M., Thambidurai P. M.* The MAFT architecture for distributed fault tolerance // Proc. IEEE Trans. on Computers. 1988. V. 37. N№ 4. P. 398–405.

77. *Walter C. J.* MAFT: an architecture for reliable fly-by-wire flight control // Proc. 8th Digital Avionics Systems Conf., 1988. Pt. 1. P. 415–421.
78. *Kiekhafer R. M.* Task reconfiguration in a distributed real-time system // Proc. Real Time Systems Sump., 1987. P. 25–32.
79. *Walter C. J., Kiekhafer R. M., Finn A. M.* MAFT: a multicomputer architecture for fault-tolerance in real-time control systems // Proc. Real Time Systems Sump., 1985. P. 133–140.
80. *Gluch D. P., Paul M. J.* Fault-tolerance in distributed digital fly-by-wire flight control systems // Proc. 7th Digital Avionics Systems Conf., 1986. P. 507–514.
81. *Myers A. F., Earls M. R., Callizo L. A.* HiMAT onboard flight computer system architecture and qualification // Proc. AIAA III Computers in Aerospace Conf., 1981. P. 41–54.
82. *Clune E., Segall Z., Siewiorek D.* Validation of fault-free behavoir of a reliable multiprocessor system FTMP: a case study // Proc. American Control Conf., 1984. V. 2. P. 1112–1120.
83. *Sitkin E., Growley C.* A real-time kernel for a 1750A-based multiprocessor // Proc. 8th Digital Avionics Systems Conf., 1988. Pt. 2. P. 748–755.
84. *Castro H. D. S., Gough M. P.* A fault-tolerant multi-transputer system for space applications // Microprocessors and Microsystems. 1991. V. 15. № 7. P. 361–367.
85. *Казьмин А. И., Менн А. А., Буяновский Д. И. и др.* Распределенная операционная система ПАРУС. М.: Ин-т проблем управления, 1987.
86. *Сырков Б. Ю.* Операционная система Chorus // Журнал д-ра Добба. 1992. № 2. С. 24–27.
87. *Tsay D. P., Liu M. T.* MIKE: a network operating system for the distributed double-loop computer network // IEEE Trans. on Software Engineering. 1983. V. SE-9. № 2. P. 143–154.
88. *Mellor P. V., Dubery J. M., Whitehead D. G.* Adapting Modula-2 for distributed systems // Software Engineering J. 1986. V. 1. № 5. P. 184–189.
89. *Менн А. А.* Распределенные операционные системы управляющих комплексов ЭВМ // АиТ. 1988. № 1. С. 3–37.
90. *Banino J. S., Fabre J. C., Guillemoni M. et al.* Some fault-tolerant aspects of the CHORUS distributed system // Proc. 5th Distributed Computer Systems Conf., 1985. P. 430–437.
91. *Banino J. S., Fabre J. C.* Distributed coupled actors: a CHORUS proposal for reliability // Proc. 3rd Distributed Computer Systems Conf., 1982. P. 128–134.
92. *Banino J. S.* Parallelism and fault-tolerance in the CHORUS // J. of Systems and Software. 1986. V. 6. № 1–2. P. 205–211.
93. *Loques O. G., Kramer J.* Flexible fault tolerance for distributed computer systems // IEE Proc. Pt. E. 1986. V. 133. № 6. P. 319–332.
94. *Knight J. C., Urquhart J. I. A.* On the implementation and use of Ada on fault-tolerant distributed systems // IEEE Trans. on Software Engineering. 1987. V. SE-13. № 5. P. 553–563.
95. *Sinha A., Das P. K., Chaudhuri A.* Checkpointing and recovery in a pipeline of transputers // Microprocessing and Microprogramming. 1992. V. 35. № 1–5. P. 141–147.
96. *Stainov R., Kalfa W.* A light weight kernel server // Microprocessing and Microprogramming. 1992. V. 35. № 1–5. P. 39–45.
97. *Burkowski F. J., Cormack G. V., Dyment J. D., Pachl J. K.* A message-based architecture for high concurrency // Proc. Hypercube Multiprocess. 1st Conf. Knoxville, Tex., 1986. P. 27–37.
98. *Василеску Ю.* Прикладное программирование на языке Ада. М.: Мир, 1990.
99. *Pountain D.* Occam II // BYTE. 1989. V. 14. № 10. P. 279–284.
100. *Vajda F.* Concurrent systems, programming primitives and languages: a comparative study // Microprocessing and Microprogramming. 1986. V. 18. № 1–5. P. 185–194.
101. *Баррон Й., Кэвил П., Мэй Д., Вильсон П.* Транспьютер с быстродействием 5 млн. операций/с и более // Электроника. 1983. № 23. С. 26–35.
102. *Barron J. M.* The transputer and Occam // Proc. Information Processing-86 IFIP Conf., 1986. P. 259–265.

103. *Джоунз Г.* Программирование на языке Оккам. М.: Мир, 1989.
104. *Бахтеяров С. Д.* Язык программирования Оккам. М. МНИИПУ, 1989.
105. *Краснов С. А.* Транспьютеры, транспьютерные вычислительные системы и Оккам // Вычислительные процессы и системы. Вып. 7. М.: Наука, 1990. С. 3–93.
106. *Семик В. П., Агаронян А. Л., Каменнова М. С.* Технология программирования параллельных вычислительных систем на базе транспьютерных сетей // Итоги науки и техники. Сер. Техническая кибернетика. М.: ВИНИТИ, 1990. Т. 30. С. 3–50.
107. *Красковский А. Ю., Чепин Е. В.* Транспьютеры: технические характеристики и опыт использования // Зарубежная радиоэлектроника. 1991. № 4. С. 43–50.
108. *Hull M. E. C., Zarea-Aliabadi A.* Real-time system implementation – the transputer and Occam alternative // Microprocessing and Microprogramming. 1989. V. 26. № 2. P. 77–84.

Поступила в редакцию 28.11.94