Лекция. **КОИУ повышенной надежности**

1. Пути повышения надежности
2. Методы и средства отказоустойчивости

1. Пути повышения надежности

В ходе эволюционного развития вычислительной техники перед разработчиками автоматизированных систем и их элементов постоянно стояли и стоят две задачи. Первая связана с максимизацией производительности вычислительных средств, вторая – с обеспечением их высокой надежности. Решение первой задачи сопряжено с повышением быстродействия элементной базы, распараллеливанием вычислительного процесса и созданием эффективных алгоритмов и программ. Решение второй задачи предполагает работу в двух основных направлениях.

Сущность первого направления состоит в предотвращении возникновения сбоев и отказов. Конечные цели этого направления достигаются: повышением технологического уровня изготовления компонентов системы, повышением степени комфортности окружающей среды, а также минимизацией ошибок разработчиков и обслуживающего персонала. Улучшению надежностных характеристик компонентов и всей системы в целом способствуют: входной контроль комплектующих элементов, повышение степени их интеграции, а также применение эффективных методов рассеивания тепловой энергии элементов. Однако данный подход имеет свои естественные ограничения технологического и экономического характера. Подобного рода ограничения встают перед разработчиками и обслуживающим персоналом при обеспечении комфортных условий функционирования автоматизированных систем. Полное устранение ошибок, вносимых при создании и эксплуатации, принципиально невозможно в силу психофизических особенностей человека.

При реализации второго направления в достижении высокой степени надежности априорно допускается возникновение сбоев и отказов, но предполагается использование эффективных методов устранения их последствий. В этом случае принято говорить о невосприимчивости вычислительного процесса к различного рода нарушениям. Системы, в основе которых лежит использование подобного подхода, получили название *отказоустойчивых вычислительных систем*.

Под **отказоустойчивостью** вычислительной системы понимают ее свойство, позволяющее продолжать выполнение заданных программой действий после возникновения одного или нескольких сбоев и отказов ее компонентов. К *отказам* относят события, заключающиеся в нарушении работоспособности каких-либо компонентов, а *сбоями* называют самоустраняющиеся отказы компонентов, приводящие к кратковременной утрате ими работоспособности. Проявление сбоя или отказа при решении задач называют *ошибкой*.

Применение отказоустойчивых систем значительно повышает оперативность решения задач пользователей вследствие уменьшения перерывов в ходе обработки информации. Подобные системы представляют определенный инте-

рес, потому что позволяют непрерывно или, по крайней мере, в течение гораздо большего времени, чем обычные, компьютеры вести расчеты или управлять объектами, так как с высокой вероятностью гарантируют обнаружение и локализацию ошибок, а затем оперативное восстановление данных и вычислительного процесса.

Последовательность этапов восстановления вычислительного процесса выглядит следующим образом. Изначально система находится в работоспособном состоянии и ведет обработку данных. В некоторый случайный момент времени возникает ошибка. Спустя определенное время ошибка выявляется, а затем локализуется место ее возникновения. После этого происходит реконфигурация компьютерной системы с целью изоляции отказавшего компонента. Далее осуществляется восстановление потерянной вследствие ошибки информации. Заключительным этапом является восстановление вычислительного процесса в целом. В результате указанных действий вычислительная система переходит в работоспособное состояние и продолжает выполнение прерванных работ.

Классификация ошибок вычислительного процесса

Рассмотрение средств и методов выявления ошибок, устранения их последствий, а также восстановления вычислительного процесса предполагает исследование характера возможных ошибок. Представляется целесообразной следующая *классификация ошибок*, возникающих в работе компьютерных систем.

По времени действия различают *постоянные* ошибки, связанные с отказом компонента системы, *временные или перемежающиеся* ошибки, возникающие в результате неустойчивой работы компонента, и *случайные* ошибки, связанные со случайными воздействиями среды на работу системы.

По количественному признаку ошибки бывают *одиночные*, то есть вызванным сбоем или отказом одного компонента системы, и *множественные* – вызванные сбоем или отказом одного (нескольких) компонентов и проявляющиеся в нескольких областях системы одновременно.

По характеру проявления ошибки делятся на *ошибки обращения*, то есть вызванные обращением к запрещенному или несуществующему состоянию системы, *ошибки сравнения* – следствие неверного результата сравнения двух и более величин, *ошибки задержки* – следствие несовпадения времени выполнения идентичных операций и *программные ошибки*, возникающие в результате неправильной работы программ (например, зацикливание).

2. Методы и средства отказоустойчивости

а) *методы и средства контроля вычислительного процесса*

В основе разнообразных методов контроля работы ОУВС лежит *принцип избыточности*, которая может быть двух видов: временная избыточность и избыточность программного и/или аппаратного обеспечения.

Временная избыточность реализуется программно путем проведения некоторых дополнительных вычислений, необходимых для сравнения результатов отдельных операций или вычислений. Совпадение результатов является основанием для продолжения вычислительного процесса.

Избыточность программного и/или аппаратного обеспечения предусматривает дополнительные средства, используемые для выполнения контрольных операций и сравнения их с результатами основных.

Выделяют следующие *уровни выявления ошибок*: сигнальный, тестовый и функциональный.

Средства, применяемые на *сигнальном уровне*, реализуются в виде схем с самоконтролем, которые позволяют обнаружить ошибку непосредственно после ее возникновения. Недостаток использования подобных средств обусловлен сложностью оснащения всех компонентов самоконтролирующимися схемами, что влечет значительное увеличение количества избыточного оборудования. Кроме того, наблюдается определенная физическая зависимость между контролируемыми блоками и средствами выявления ошибок. Поэтому часто отказ контролируемого элемента влечет за собой и отказ средства выявления ошибок. Тем не менее, несомненным достоинством является эффективность использования средств данного уровня. Это обусловлено высокой оперативностью обнаружения ошибки и точной локализацией места ее возникновения. К основным группам средств обнаружения ошибок на сигнальном уровне следует отнести: *схемы, использующие специальные коды*, *схемы с дублированием*, *схемы с голосованием* (мажоритарные схемы). Наибольшего внимания заслуживает использование схем с дублированием, так как при их относительной простоте происходит почти полное выявление ошибок. Редкое исключение составляют случаи, когда совпадают оба ошибочных результата.

Средства *тестирования* предназначены для генерации определенных наборов входных сигналов, называемых тестами, при подаче которых все возможные нарушения работоспособности компонентов вызывали бы ошибочные выходные сигналы. Эти ошибки обнаруживаются путем сравнения значений выходных сигналов с эталонными. Средства данного уровня в состоянии выявлять постоянные ошибки и отчасти временные. Случайные ошибки с их помощью не обнаруживаются.

Используются разнообразные *методы тестирования*, которые можно классифицировать следующим образом.

По времени проведения тестирования. Прогон тестов может выполняться *в ходе работы системы*. При этом тестированию подвергаются незадействованные в данное время подсистемы одновременно с основной работой остальных подсистем. Альтернативой является выполнение тестов *в перерывах*, например, между сменами.

По периодичности тестирования. В соответствии с указанным признаком можно выделить *периодические* прогоны тестов, проводимые в строго определенные моменты, и *непериодические*, выполняемые, например, по окончании выполнения очередного задания.

По продолжительности тестирования. Тестовые проверки могут проводиться *в течение фиксированного отрезка времени* или неопределенно долго *до локализации ошибки*. Последнее характерно при выполнении тестов в предположении, что ошибка неслучайна и ее проявление было обнаружено средствами другого уровня.

По местонахождению средств тестирования. Различают *внутреннее* и *внешнее* тестирование. В первом случае тестовые проверки осуществляются основными средствами, как правило, процессорами системы, во втором - специальными сервисными процессорами.

По способу организации тестирования. По данному признаку методы делятся на *поочередное* и *одновременное* (параллельное) тестирование компонентов системы.

По способу проведения тестирования. Возможно *самотестирование компонентов*, тестирование *соседних компонентов* и тестирование *любого компонента любым процессором*.

Средства *функционального уровня* выявления ошибок предназначены для предотвращения нежелательных действий и выявления ошибочной информации. После отказа какого-либо компонента ошибка может быстро распространяться в среде вычислительной системы и создавать эффект "ускорения ошибки" или "снежного кома". На предотвращение подобного явления и направлено использование средств функционального уровня, которые представляют собой некие "барьеры" вокруг верных результатов вычислений и правильных функциональных действий системы.

Методы установления барьеров в среде вычислительной системы могут быть различными. Сущность установления *временного барьера* состоит в отслеживании времени выполнения задания. Если задание не завершено в определенное время, то полагается, что произошла ошибка, и ее последствия могут нарушить весь ход вычислительного процесса. В этом случае задание принудительно снимается с выполнения. Установление *пространственного барьера* предполагает обеспечение контроля над распространением информации. Примером может служить следующая ситуация. Если текущей программе выделена определенная область памяти, то адресование к ячейкам, не принадлежащим данному полю, служит признаком ошибки вычислений. Ограничение распространения ошибки замкнутой областью упрощает процесс восстановления системы, так как сохраняется большое количество информации, не подверженной воздействию ошибок.

Другую группу методов осуществления контроля на функциональном уровне составляют методы подтверждения результатов вычислений. Проведением дополнительных вычислений либо подтверждаются основные результаты, либо в последних выявляются ошибки. Наиболее часто используются методы контрольных сумм и контрольных функций.

Для построения отказоустойчивых систем не вполне достаточно использования средств выявления ошибок только одного уровня. Лишь комплексное использование разнообразных средств и методов может дать требуемый результат.

б) *методы и средства устранения последствий ошибок*

После обнаружения и локализации ошибки необходимо устранить ее последствия. К числу простейших методов можно отнести повторение вычислений. Однако он позволяет устранять только ошибки, являющиеся результатами сбоев, и требует существенных затрат времени. Следующие два метода устраняют названные недостатки.

Метод маскирования ошибочных действий. Суть его состоит в том, что избыточная информация скрывает действие ошибок. Это достигается особенностями схемных решений и организации вычислительного процесса. Используются статистические средства устранения последствий ошибок, так называемые *средства маскирования*. По принципу действия их можно разделить на следующие основные группы: корректирующие коды и схемы с голосованием. В последних при организации вычислений используется нечетное число устройств для выполнения идентичных операций. Большинством голосов определяется правильный набор выходных сигналов.

Метод реконфигурации компьютерной системы. Он заключается в изменении состава вычислительных средств и/или способа их взаимодействия. Естественно, что реконфигурация проводится после обнаружения и локализации ошибки. Различают статическую и динамическую реконфигурации.

Статическая реконфигурация осуществляется отключением неисправных компонентов. Вся вычислительная система при этом делится на две части: активную и пассивную. Первая включает компоненты, непосредственно участвующие в обработке данных, вторая же охватывает неработоспособные компоненты, отключенные в ходе реконфигурации.

Динамическая реконфигурация по принципу ее проведения может представлять собой замещение компонентов (поддержка запасом), дублирование, постепенную деградацию системы (снижение вычислительных способностей). Здесь следует заметить, что если система имеет маскирующую избыточность, то изоляцию отказавших компонентов можно отложить на некоторое число отказов до тех пор, пока число отказавших элементов не станет угрожающим в смысле возникновения немаскированной ошибки.

в) *методы и средства восстановления вычислительного процесса*

Устранение последствий ошибок создает предпосылки для восстановления вычислений. Собственно процесс восстановления может проводиться на двух уровнях: аппаратном и программном.

На аппаратном уровне возможно автоматическое восстановление отказавших компонентов и ремонт. *Автоматическое восстановление* реализуется путем дополнительной реконфигурации системы. Предполагается, что в системе имеется ряд запасных элементов, благодаря которым она возвращается в работоспособное состояние. При этом производительность остается неизменной или несколько снижается. Проведение *ремонта* состоит в отключении отказавшего компонента и его восстановлении вне системы. При этом система либо продолжает функционировать с меньшей производительностью, либо переводится в режим останова до возвращения отремонтированного компонента.

В общем случае восстановление системы на аппаратном уровне не влечет за собой восстановления вычислительного процесса. Его проведение носит обеспечивающий характер и должно рассматриваться в совокупности с восстановлением *на программном уровне*. Программные методы основаны на восстановлении необходимой для продолжения работы информации о системе. В зависимости от количества и характера ошибок выделяют следующие методы восстановления.

Повторение операции дает положительный результат при возникновении случайных или временных ошибок. Развитием этого метода выступает многократное повторение, увеличивающее вероятность правильного восстановления вычислительного процесса.

Возврат к контрольной точке является разновидностью общего случая повторения. Контрольная точка – это некоторый рубежный этап вычислительного процесса, для которого зафиксированы промежуточные результаты вычислений и информация о состоянии системы, позволяющая возобновить обработку данных. При обнаружении ошибки система возвращается к контрольной точке, предшествующей моменту возникновению ошибки, и продолжает свою работу, используя эту точку в качестве исходной. Обязательным является выполнение условия, что результаты в контрольной точке не подвержены ошибкам.

Повторное выполнение программы проводится в том случае, если в системе разрушено такое количество информации, что восстановление с помощью двух предыдущих методов невозможно или нецелесообразно. Тогда все неоконченные к моменту возникновения ошибки программы подлежат выполнению с начала. Можно указать три случая повторного прогона программ. Во-первых, если последствия ошибок успели отразиться на большей части системы. Во-вторых, если возможным оказывается восстановление только некоторых частных процессов, даже при минимуме ошибок в них. И, наконец, в-третьих, если возобновление нормального функционирования системы с помо-

щью других методов сопряжено с техническими трудностями или временными затратами.

Каждый из рассмотренных методов восстановления вычислительного процесса вызывает некоторую задержку в выполнении заданий. Использование методов восстановления программного уровня требует существенно больших временных затрат, чем маскирование ошибок. Учитывая то обстоятельство, что допустимая задержка в восстановлении зависит от конкретного применения системы, определить оптимальное сочетание использования средств маскирования и программного восстановления в общем случае не представляется возможным.

Таким образом, наиболее целесообразным является сочетание различных методов и средств обнаружения и локализации ошибок, а также восстановления процесса обработки данных. Это объясняется тем, что с помощью только одного из них с достаточно высокой степенью достоверности обнаруживается лишь определенный класс ошибок. В пользу комплексного использования разнообразных методов и средств свидетельствует и то, что прогрессивное развитие элементной базы позволяет включать в состав отказоустойчивых компьютерных систем различные средства обнаружения и исправления ошибок. Также следует отметить высокую эффективность применения средств контроля и восстановления на нескольких иерархических уровнях компьютерных систем.