Киевская городская государственная администрация
Институт прикладной информатики

На правах рукописи

АЛИЕВ Алекпер Али Ага оглы

УДК 681.324

ПРОТОКОЛЫ СИНХРОНИЗАЦИИ ПРИКЛАДНЫХ ПРОЦЕССОВ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ ОБРАБОТКИ ИНФОРМАЦИИ

05.13.04 - автоматизированные системы управления и системы обработки информации

Автореферат диссертации на соискание ученой степени доктора технических наук

Киев 1996

Работа выполнена в институте про
НАН Украины

Научный консультант: доктор технических наук,
профессор Никитин А. И.

Официальные оппоненты: доктор технических наук
Аваров С. С.,
доктор физико-математических наук,
профессор Вельбицкий И. В.,
доктор технических наук,
профессор Вертузаев М. С.

Ведущая организация: Национальный Технический Университет Украины
(Киевский политехнический институт).

Защита состоится "10" апреля 1996 г. в 14[00] часов на заседании специализированного совета Д 01.64.01 в Институте прикладной информатики по адресу:
252004, г. Киев, ул. Красноармейская, 23-б.

С диссертацией можно ознакомиться в библиотеке Института прикладной информатики.

Автореферат разослан "7" марта 1996 года.

Ученый секретарь
специализированного совета Мелентьев Г. В.

Підписано до друку: 4.03.96 Формат 60x84 1/16.
Об'єм 3,05 д.а. Зам.N 27 Тираж 100 прим. Ціна
Державне комунальне поліграфічне підприємство "Тираж"
м. Київ

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В процессе широкой информатизации общества в целом, большое значение приобретает создание и развитие компьютерных сетей и построение на их основе распределенных систем обработки данных или просто распределенных систем (РС). РС предоставляет конечным пользователям несравненно более широкий набор информационных и вычислительных услуг. РС позволяют интегрировать информационные ресурсы в рамках предприятия, региона, страны и частично в международном масштабе. Особенно важна такая интеграция в решении задач управления экономикой, где управляющий орган может получать информацию любой детализации в статике и динамике и тем самым оперативно отслеживать социально-экономические процессы, протекающие в стране. Вместе с тем создание РС требуют предварительного решения многих достаточно трудных проблем.

Одной из проблем, возникающих в теории и практике РС, функционирующих в сетевой среде, является синхронизация процессов, протекающих в узлах РС и конкурирующих между собой за использование ее ресурсов. Синхронизация понимается как скоординированное корректное управление процессами, протекающими в РС как последовательно, так и параллельно. Без синхронизационных механизмов, являющихся аналогом устройства управления в компьютере, невозможно выполнение множества подзадач в узлах РС, решающих в совокупности единую крупную задачу. К подзадачам в основе которых лежит проблема синхронизации процессов в РС можно отнести задачу взаимного исключения при выделении ресурса, задачу обеспечения сериализуемости плана выполнения транзакций, задачу двухфазной фиксации транзакций, задачу определения глобального состояния РС без остановки процесса выполнения транзакций и т. п.

Другой проблемой является обеспечение отказоустойчивости РС. Отказоустойчивыми называются такие системы, которые способны продолжать корректное функционирование при возникновении разнообразных отказов их компонентов и сохранение целостности распределенной базы данных (РБД). Конкретно, при возникновении отказа одного из компьютеров (узлов) сети необходимо так организовать информационно-вычислительный процесс, чтобы за возможно более короткое время отказавший узел был обнаружен и об этом стало известно всем остальным узлам сети для определения всех возможных отрицательных последствий отказа, произошедших до того момента, когда отказавший узел был подвергнут временной информационной изоляции (отторжению). Эта

естественная, на первый взгляд, последовательность действий по обеспечению отказоустойчивости сети достаточно сложна в реализации. Действительно, непросто ответить на вопрос, какой из вполне равноправных узлов должен взять на себя инициативу для инициации действий диагностирования, временной изоляции, а затем "восстановления в правах" отказавшего узла.

Одной из фундаментальных проблем перспективных РС является проблема определения глобального состояния РВД при выполнении параллельных транзакций и при выходе из строя одного или более узлов РС. Эта проблема не достаточно исследована и разработана. Идея определения глобального состояния РС лежит в основе решения многих задач, одной из которых является задача формирования глобальной контрольной точки, которая служит не только для рестарта РС при отказах оборудования, но и является основой решения некоторых специфических прикладных задач в РС, например, для ревизии счетов распределенной банковской системы, переписи населения и др.

Перечисленные трудности не могут, однако, не только остановить, но и сколько-нибудь затормозить широкое всеохватывающее распространение РС - одного из самых значительных достижений научно-технического прогресса конца двадцатого столетия.

Указанные обстоятельства обусловливают актуальность тематики данной работы.

Настоящая диссертационная работа направлена на решение проблемы взаимодействия прикладных процессов в РС и корректного упорядочения (синхронизации) этих процессов.

Цель работы и основные задачи. Цель диссертационной работы состоит в развитии теории взаимодействия и синхронизации прикладных процессов в распределенных системах обработки информации, построенных на основе компьютерных сетей и внедрении полученных результатов в практике РС. Исходя из поставленной цели в работе решаются следующие основные задачи:

- исследование общих принципов распределенного плана выполнения транзакций над РВД и построение протоколов, позволяющих предупредить возникновение таких недопустимых ситуаций как несериализуемось плана и тупик;

- разработка протоколов фиксаций транзакций в условиях параллельного их выполнения, повышающих производительность РВД и гарантирующих при этом сохранение ее целостности;

- исследование глобального состояния РС и систематизация существующих протоколов, фиксирующих глобальное состояние РВД;

- разработка эффективного метода отката-восстановления взаимодействующих процессов, позволяющего предотвратить "эффект домино;

- разработка рациональных протоколов формирования глобальных контрольных точек РВД, исходя из которых можно восстановить корректное функционирование системы после отказа некоторого узла сети;

- внедрение полученных теоретических результатов в РС.

Методы исследования. Разработка и исследование синхронизационных протоколов основаны на применении теории графов, конечных автоматов, надежности, баз данных, вычислительных систем.

Научная новизна. В работе впервые сделана попытка охватывать все теоретические вопросы реализации синхронизационных процессов в распределенных системах обработки данных. Полученные теоретические и прикладные результаты, позволяют обобщить и решить крупную научную проблему взаимодействия прикладных процессов и достижения консенсуса в децентрализованных РС.

В работе получена совокупность следующих результатов, характеризующих научную новизну исследования:

- Обобщены принципы построения распределенного плана выполнения транзакций и свойства сериализуемости этого плана. Построен протокол, позволяющий заранее предупредить возникновение несериализуемого распределенного плана.

- Разработан и обоснован ряд протоколов, решающих задачу сериализуемости и основанных на понятии глобального лэмпортова времени в РС, а также на алгоритме построения растущего дерева зависимости транзакций.

- Предложена единая иерархическая структура механизмов обеспечения отказоустойчивости РС и построены протоколы завершения (фиксации) транзакций в условиях параллельного их выполнения, позволяющие сохранить целостность РБД как при исправной сети, так и при отказе отдельных ее узлов.

- Предложены и обоснованы принципы использования механизма отката-восстановления как базового программного средства обеспечения отказоустойчивости РС, обладающего достаточной универсальностью и высокой технологичностью использования. Проведено исследование этого механизма с применением аппарата контрольных точек и предложены методы

отката-восстановления, позволяющие предотвратить "эффект домино".

- Предложено обобщение протокола "моментального снимка", фиксирующего глобальное состояние сети по Лэмпорту на основе "перекрашивания" узлов.

- Разработаны рациональные протоколы формирования глобальных контрольных точек РБД, к которым надлежит вернуть систему после отказа и восстановления технических средств сети, чтобы затем корректно рестартовать и продолжить процесс транзактной обработки. Эти протоколы сводят к минимуму число откатываемых транзакций и уменьшают простой РС при формировании глобальных контрольных точек и, следовательно, повышают производительность системы управления транзакциями, а также отличаются наличием логических связей с протоколом двухфазной фиксации транзакций.

Практическая ценность и реализация результатов работы. Разработанные в диссертационной работе методы и протоколы направлены на развитие теории взаимодействия прикладных процессов в распределенных системах обработки данных, построенных на основе компьютерных сетей. В прикладном плане работа направлена на решение следующих задач:

- создание сетевых протоколов, обеспечивающих сохранение целостности РБД;

- управление работой компьютерных сетей в реальном масштабе времени;

- повышение отказоустойчивости РС;

- повышение производительности системы управления транзакциями.

Основные результаты диссертационной работы были использованы и внедрены в распределенной автоматизированной системе обработки информации (АСОИ) АН Азербайджана.

Апробация работы. Основные результаты диссертационной работы докладывались и обсуждались на 5-ой Всесоюзной конференции "Вычислительные сети коммутации пакетов" (г. Рига, 1987), 1-ой Всесоюзной научно-технической конференции "Методы анализа надежности программного обеспечения вычислительных систем реального времени на основе моделей нечеткой логики и качественных описаний" (г. Киев, 1987), 2-й Всесоюзной научно-технической конференции "Живучесть и реконфигурация информационно-вычислительных и управляющих систем" (г. Алушта, 1988), Конференции молодых ученых и специалистов Института кибернетики имени В. М. Глушкова АН Украины (г. Киев, 1988), 6-й

Всесоюзной конференции "Вычислительные сети коммутации пакетов" (г. Рига, 1989), республиканской научно-практической конференции молодых ученых и специалистов (г. Баку, 1990), 1-ой Всесоюзном научно-практическом семинаре (г. Харьков, 1990), семинарах научного совета АН Украины по проблеме "Кибернетика", семинарах Научно-производственного центра при Бакинском Государственном Университета.

Публикации. По результатам диссертации опубликованы 20 печатных работ и монография, выполненные автором самостоятельно и в соавторстве.

Структура и объем работы. Диссертационная работа состоит из введения, пяти глав, заключения, списка литературы, приложения и содержит 258 страниц машинописного текста. Список литературы включает 216 наименований. Работа содержит 36 рисунков и 10 таблиц.

Во введении обосновывается актуальность исследуемой в диссертационной работе темы, формулируются цель и задачи исследования, дается краткая характеристика работы.

В первой главе рассматриваются проблемы развития РС обработки информации в компьютерных сетях. Приводятся основные сведения о компьютерной сети так, как она понимается в эталонной модели взаимосвязи открытых систем, предложенной Международной организацией стандартов, что позволяет практически полностью отвлечься от технических деталей сети и вести изложение на логическом уровне.

В работе принята модель РС, которую можно описать следующим набором правил (предположений):

М1. РС является совокупностью узлов и системы передачи данных, связывающей любую пару узлов.

М2. В узлах РС хранится и обрабатывается прикладными процессами информация, организованная в виде баз данных (БД) под контролем локальных систем управления базами данных (СУБД).

М3. Прикладные процессы взаимодействуют друг с другом только путем обмена сообщениями, которые могут либо инициировать содержательную обработку информации, либо выполнять чисто управленческие функции. Первый тип сообщения обозначим M(i,j), а второй - m(i,j), где i,j - номера узлов источника и адресата сообщения соответственно.

М4. Система передачи данных (сообщений) надежна, т.е. сообщения не исчезают и не возникают самопроизвольно, все узлы доступны для сообщений.

Каждый прикладной процесс имеет возможность доступа к любой БД,

функционирующей в РС независимо от того, какая логическая схема соответствует той или иной БД. Множество БД, связанных между собой в процессе решения задач, образуют в узлах РВД. РС не зависит от изменений как аппаратного оборудования, так и операционных систем, функционирующих в узлах.

Несмотря на простоту введенная модель позволяет решать многие теоретические задачи управления, достаточно близкие к задачам, возникающим в реальных практически важных РС.

Большинство задач синхронизации, которые описаны в последующих главах, ставятся и решаются в соответствии с этой моделью, только в отдельных случаях модель несколько модифицируется.

Далее приводятся примеры прикладных РС, их потенциальные преимущества и трудности на пути их реализации. Основным достоинством РС является то, что они приближают точки доступа к информационно-вычислительным ресурсам непосредственно к конечному пользователю, к его рабочему месту, к его месту проживания, в пределе к любому месту, где бы он ни находился в данный момент. Это существенно повышает психологический комфорт пользователя и оперативность его взаимодействия с информацией любого содержания, представленной в любой практически полезной форме. Уже сейчас можно прогнозировать значительные социальные последствия такого приближения: повысится процент работающих на дому, разгрузятся такие сервисные службы, как почта, периодическая и справочная печать, торговля, продажа билетов на транспорт.

Особое значение приобретает возможность автоматизации процессов всякого рода платежей и в первую очередь в банковской сфере. На базе РС можно построить систему безналичного расчета, охватывающую всю страну. Такая система будет работать эффективно только тогда, когда денежные операции смогут проводиться в любой точке территории в любое время суток. При этом должна быть создана надежная система аутентификации информации, гарантирующая безошибочное распознование личности, производящей денежную операцию.

РС способны органически вписаться в социальные и хозяйственные механизмы страны. Органичность понимается как естественное дополнение (или естественная замена) автоматическими компонентами ручных или полуавтоматических компонентов, не требующее каких-либо дополнительных усилий со стороны конечного пользователя, т.е. не нарушающее привычный распорядок работы. Это свойство образно характеризуется как

"прозрачность" новых компонентов для конечного пользователя. "Прозрачность" является одним из основных требований ко многим компонентам вычислительной техники и достигается не всегда просто, но, будучи доступной, быстро окупает свою разработку улучшением эргатических свойств системы.

РС могут широко использоваться в ходе выборов, референдумов, а также в обычной обстановке для передачи информации от любого гражданина в директивные органы. Протоколы РС можно разработать таким образом, что письма трудящихся не будут "оседать" в канцеляриях, а попадут нужному адресату в режиме полной конфиденциальности. Трудно переоценить вклад подобных РС в развитие демократизации нашего общества. Перечисленные выше преимущества РС требуют предварительного решения многих достаточно трудных проблем.

Одной из проблем является синхронизация процессов, протекающих в узлах РС. В работе определяется в общем виде задача синхронизации, как построение функции упорядочения событий в системе, в соответствии с некоторыми содержательными критериями, специфическими для данной задачи. Цель построения функции упорядочения событий состоит в том, что сообщение о событии-причине должно во всех прикладных процессах отрабатываться раньше, чем сообщение о событии-следствии. Сообщения о параллельных событиях могут отрабатываться в произвольном порядке. Чтобы достичь этой цели, необходимо снабдить каждое сообщение уникальной меткой - значением функции упорядочения. Функция упорядочения называется также лэмпортовым временем.

Следует отметить, что вводя модель РС набором предположений М1-М4, по умолчанию предполагается, что процессоры, функционирующие в узлах РС, надежны, т.е. не подвержены ни сбоям, ни отказам. Для корректности дальнейших рассуждений выпишем это предположение в явном виде:

М5. Процессоры, функционирующие в узлах РС, надежны.

Большинство задач синхронизации может решаться на основе лэмпортова времени, т.е. без использования часов, отсчитывающих реальное время, или интервальных таймеров. Но введение реального времени дает возможность отказаться от достаточно жесткого предположения в модели РС, а именно - предположения М5. Это дает возможность значительно приблизить модель к реальным системам, так как абсолютно надежных процессоров пока еще не существуют. Итак, вместо предположения М5 можно принять предположение М6 :

М6. Процесссры РС могут отказывать в случайные моменты времени.

По этому поводу необходимо сделать одно замечание. В третьей главе мы откажемся от предположения МБ, но методически целесообразно вначале рассмотреть именно РС с надежными узлами, что мы и делаем во второй главе.

Здесь уместно упомянуть, что синхронизационные механизмы функционируют не только на прикладном уровне ВОС-МОС, но и на всех более низких уровнях. Однако именно протоколы управления РБД наиболее интересны потому, что в них превалируют множественные запросы, т.е. запросы к нескольким локальным БД одновременно.

Одной из простых задач, в основе которых лежит проблема синхронизации процессов в РС, является задача взаимного исключения при выделении ресурса. Для РС задача взаимного исключения формируется следующим образом.

Пусть существует n различных независимых параллельных процессов, функционирующих в РС, которые могут потенциально претендовать на один и тот же разделяемый ресурс. Как и выше, будем без потери общности отождествлять эти процессы с узлами РС. Требуется построить протокол, который упорядочивает доступ к разделяемому ресурсу так, чтобы в каждый момент времени не более чем один узел производил действия над разделяемым ресурсом, не опасаясь конкурентных действий других узлов. Будем говорить, что узел, владеющий разделяемым ресурсом, находится в состоянии mutex (mutual exclusion) или в критической секции. Естественно, что могут существовать временные интервалы, внутри которых ни один узел не находится в состоянии mutex.

Одна из идей, на которых базируются алгоритмы взаимного исключения, состоит в том, что в РС существует и перемещается уникальный управляющий маркер (control token). Только тот узел, который обладает в некоторый момент времени этим маркером, может перейти в состояние mutex. По завершении работы узел выходит из этого состояния. Очевидно, что, поскольку маркер уникален, один и только один узел в некоторый момент времени может находиться в состоянии mutex.

Другой широко известный в литературе механизм, нашедший применение для решения задачи взаимного исключения, основан на введении временных меток. В работе рассмотрен протокол взаимного исключения, основанный на идее временных меток. Он наиболее прост и приведен с тем, чтобы продемонстрировать на примере типовые утверждения, которые как правило, должны быть доказаны при решении практически всех задач синхронизации.

Во второй главе рассматривается модель РС, заданная предположениями М1-М5. В настоящее время общепринятым методом обработки информации в РС является транзактная обработка. В множестве узлов S в каждый момент времени функционируют два подмножества:

{TM} - подмножество узлов, инициирующих транзакции;

{DM} - подмножество узлов, обрабатывающих транзакции.

Субтранзакции одной и той же транзакции, а также разных транзакций в различных узлах DM могут выполняться параллельно. Ввиду случайности и независимости процесса инициации транзакций узлами TM в узлах DM могут образовываться очереди субтранзакций, ждущих обработки. Совокупность всех очередей (из которых некоторые могут быть пустыми) назовем распределенным планом выполнения субтранзакций - DSO (Distributed Subtransaction Ordering).

Координация параллельной обработки транзакций в рамках всей РС проводится системой управления транзакциями (СУТ), имеющий своих агентов в каждом из узлов TM и DM. Основными требованиями, которые предъявляются к СУТ, являются обеспечение: принципа атомарности транзакций; сериализуемости DSO; статистической справедливости по отношению ко всем инициированным транзакциям; максимальной производительности РС за счет высокой степени параллелизма выполнения субтранзакций в различных узлах; отказоустойчивости РС.

В работе рассматривается условие обеспечения сериализуемости DSO. Строится так называемый Q - граф, соответсвующий очередям субтранзакций в DM, т.е. множеством вершин этого графа будет множество всех субтранзакций, стоящих в очереди. Дугами соединяются те пары вершин, которые попали в общую очередь и являются соседями в ней; дуга направлена от некоторой субтранзакции, стоящей в очереди, к субтранзакции, стоящей непосредственно перед ней. Максимально Q - граф имеет столько компонент связности, сколько узлов DM имеется в РС. Каждая компонента связности Q - графа называется локальным планом выполнения субтранзакций для соответствующего узла DM. Далее склеивая вершины Q - графа, имеющие одинаковые нижние индексы, строится так называемый D - граф (граф зависимостей).

D - граф является в общем случае мультиграфом и число его вершин равно числу узлов TM. Признаком сериализуемоси DSO является ацикличность соответствующего D - графа. Некоторый DSO - называется сериализуемым, если он эквивалентен по крайней мере одному последовательному (сериальному) плану.

Важность класса сериализуемых DSO вытекает из следующих рассуждений. Каждая транзакция программируется так, чтобы, начав работать над целостной РБД, она оставляла после себя также целостную РБД. Из этого непосредственно вытекает, что любой последовательный план также не нарушает целостности РБД. К понятию целостности РБД мы вернемся ниже, здесь же отметим только, что нарушение целостности РБД - это событие, равносильное информационной катастрофе и ни в коем случае не должно быть допущено. Учитывая тот факт, что сериализуемый DSO приводит РБД к тому же состоянию, что и некоторый последовательный, можно утверждать, что сериализуемый план не нарушает целостности РБД.

Для доказательства сериализуемости DSO достаточно доказать его эквивалентность хотя бы одному последовательному плану. Наличие нескольких последовательных планов, эквивалентных исходному, ничего нового не дает. При этом следует отметить, что различные последовательные планы, эквивалентные исходному, приводят РБД к одному и тому же конечному состоянию через различные промежуточные. При одном и том же исходном множестве транзакций, но при различных транспортных задержках (в значительной мере случайного характера) могут возникать различные сериализуемые DSO и, как следствие, различные соответствующие последовательные планы, приводящие в общем случае к различным конечным состояниям РБД. Нельзя судить, какое из этих состояний правильное, а какое неправильное, можно утверждать лишь одно: все они сохраняют целостность РБД.

Задача СУТ ограничивается тем, что обеспечивает сериализуемость DSO. Эта задача прямо или косвенно сводится к проверке D - графа на его ацикличность. При большом числе узлов РС такая проверка достаточно сложна, поэтому стараются сразу при постановке субтранзакций в очереди обеспечить сериализуемость DSO, т.е. устранить возможность образования цикла в D - графе.

В работе описан протокол, обеспечивающий атомарность транзакции, который назовем базовым, или B-протоколом. Точнее, B-протокол представляет собой лишь основу для построения полных протоколов. Ниже и в последующих главах на основе детализации и развития B-протокола будут сформированы полные протоколы различной функциональной направленности и для другой модели сети. B-протокол разработан для модели сети M1-M5, т.е. для надежной сети. Для наглядного описания протокола вводится формализованный язык, близкий по сути к языку

описания протокола конечных автоматов, т.е. основывающийся на двух элементах: состояниях транзакции и порожденных ею субтранзакций, а также событиях, вызывающих переходы между этими состояниями.

В этой же главе предложены несколько методов индикации несериализуемости DSO и обеспечения его сериализуемости.

Первый метод - протокол блокировки - является индикационным, т.е. применение его обеспечивает обнаружение несериализуемого DSO. Если таковой обнаружен, то производится откат одной или нескольких транзакций.

Разработан протокол, обеспечивающих атомарность транзакций, а также не допускающих выполнения несериализуемого DSO, так называемый протокол двухфазной блокировки (2PL-протокол). В случае возникновения в D-графе цикла DSO попадает в тупиковое состояние, что и указывает на несериализуемость DSO. Никаких механизмов по предупреждению тупикового состояния нет, т.е. при попадании в тупиковое состояние необходимо включить механизмы по выходу из него. Для построения 2PL-протокола использован за основу B-протокол.

Структура транзакции и способ управления ею, удовлетворяющие принципу 2PL, характерны тем, что в процессе выполнения транзакции можно выделить два периода: роста и деградации. В первом число ресурсов, захваченных транзакцией, монотонно возрастает (второй период начинается в тот момент, когда транзакция освободила хотя бы один ресурс). Во втором - монотонно убывает, т.е. она освобождает уже отработанные ресурсы, но не захватывает ни одного нового.

В работе рассмотрены централизованное и децентрализованное блокирование. Доказано, что если в потоке транзакций все транзакции удовлетворяют принцип 2PL и DSO выполняется, то последний сериализуем.

Второй метод - протокол временных меток - является превентивным, т.е. не допускает образования несериализуемого DSO, вновь поступающие транзакции, которые могут вызвать несериализуемость, откатываются.

В главе рассмотрены два протокола управления транзакциями, обеспечивающие сериализуемость последних: базовый и консервативный протоколы временных меток. Они основаны на том, что каждая субтранзакция сопровождается так называемой временной меткой ts, позволяющей каждой отдельной субтранзакции определить, грозит ли постановка ее в очередь ресурсу формированию цикла в D - графе. Если такая угроза существует, то соответствующая транзакция откатывается, с тем чтобы через некоторое время стартовать вновь. Временные метки,

генерируемые одним и тем же узлом отличаются друг от друга значением локального времени; временные метки, генерируемые различными узлами, по крайней мере, номерами узлов. Из этого следует уникальность каждой временной метки в рамках всей РС.

Сущность методов, основанных на временных метках, состоит в том, что транзакции должны обрабатываться в порядке возрастания их временных меток. Среди этих методов наиболее известным является так называемый базовый ts - протокол, основанный на том, что ресурс (база данных) в каждом узле хранит метку последнего сообщения вызвавшего изменения в базе.

Предложен протокол обеспечивающий сериализуемость DSO. Доказаны следующие утверждения.

Утверждение. При базовом методе временных меток тупиковая ситуация невозможна.

Утверждение. Протокол базового метода временных меток результативен и справедлив.

В отличие от базового ts - протокола консервативный метод относится к классу методов с ожиданием, в котором исключаются рестарты. В соответствии с этим методом в каждом узле DM функционирует планировщик, основная функция которого заключается в ведении упорядоченной очереди по возрастанию временных меток субтранзакций, поступающих из разных TM. На выполнение передается субтранзакция с минимальной временной меткой.

Третий метод - оптимистический подход - является превентивным, его преимущества в том, что есть надежда, что конфликты в РС являются исключением, т.е. происходят достаточно редко и следовательно откату подлежат незначительное число транзакций.

В оптимистическом подходе выполнению подлежат все транзакции, которые были инициированы. Однако транзакции считывают информацию из БД, обрабатывают ее и записывают результат в личное пространство. После завершения этих действий производится проверка наличия конфликтов данной транзакции с другими транзакциями, выполняющими или проводящими аналогичную проверку. Такая проверка называется ратификацией. Если ратификация завершилась успешно, т.е. конфликт между транзакциями отсутствует, то транзакции фиксируются, в противном случае результаты выполнения всех конфликтующих транзакций аннулируются, а сами транзакции подлежат последующему рестарту.

Различают локальную и глобальную ратификацию. Она проводится

сначала в узле-инициаторе данной транзакции, и эта стадия называется локальной ратификацией. Если она завершается успешно, то на следующей стадии производится глобальная ратификация. Цель последней заключается в выявлении глобальных конфликтов ратифицируемой транзакции.

В работе в рамках оптимистического подхода предложен протокол управления транзакциями, который применяет механизм временных меток на многомерный случай для проведения фазы глобальной ратификации. Данный протокол обобщает следующее утверждение.

Утверждение. Пусть узлы-инициаторы транзакций входят в глобальную критическую секцию. Независимо от того, какой из узлов окажется победителем, любой из них устанавливает одно и то же отношение линейного порядка.

Четвертый метод - прямой (или протокол растущего дерева) - также является превентивным, его преимущество в том, что откатываются только те транзакции, которые действительно (а не потенциально) образуют несериализуемый DSO. Этот метод основан на идее включения в СУТ механизма прямой проверки некоторого множества субтранзакций на отсутствие цикла в соответствующем этому множеству D - графе.

Назовем это множество "испытуемым". Если такой цикл отсутствует и возникновение его в будущем также невозможно, проверенное множество считается ратифицированным и субтранзакции, принадлежащие этому множеству, могут выполняться в том порядке, в котором они стоят в очередях. При наличии цикла одна из транзакций, инцидентная ему, откатывается и через некоторое время стартует вновь. Алгоритмы поиска цикла, точнее принятие решения, о том, существует цикл или нет, могут быть различными.

В работе в рамках прямого метода предложен один из алгоритмов (протоколов) ратификации, который будем называть протоколом растущего дерева. Он основан на дискретизации процесса инициации транзакций и построении в каждом ТМ дерева зависимостей между субтранзакциями, которое в процессе роста определяет наличие или отсутствие цикла в D - графе. Процесс инициации дискретизируется для того, чтобы циклы в D - графе не образовывались за конечное (на практике достаточно малое) время. Дискретизация заключается в следующем. Временная ось всех ТМ делится на равные отрезки, называемые интервалами дискретности. Начальные отрезки каждого интервала дискретности называются интервалами инициации.

В работе доказывается следующее утверждение.

Утверждение. Протокол результативен за r интервалов дискретности.

В третьей главе рассматриваются проблемы обеспечения отказоустойчивости РС, узлы которой могут отказывать в случайные моменты времени, т.е. принимается модель РС М1-М4 и М5. Целью обеспечения отказоустойчивости является сохранение системой возможности требуемого обслуживания при наличии в ней отказов.

Наиболее характерными классами компьютерных систем, для которых ведутся интенсивные поисковые исследования и инженерные разработки с целью повышения их отказоустойчивости, являются системы управления техническими и технологическими объектами, а также системы массового сервиса, как правило, реализующие транзактную технологию. Для систем управления, как правило, удается ввести функцию убытка, отражающую степень ухудшения управления в зависимости от времени простоя компьютерной системы.

В системах с транзактной обработкой в качестве функции убытка также рассматривается дискретная функция, характеризующая факт нарушения или ненарушения целостности РБД. Такая постановка вопроса, является в настоящее время особенно актуальной в связи со всеобщей компьютеризацией общества и катастрофическими последствиями нарушения целостности больших РБД.

Свойством отказоустойчивости обладают многие технические системы, но компьютерные являются в данном случае наиболее характерными, так как они способны адаптироваться к изменяющимся условиям, т.е. перестраивать алгоритмы своего функционирования в широком диапазоне.

Среди компьютерных систем свойство отказоустойчивости в наибольшей степени присуще, во всяком случае потенциально, распределенным (вычислительным) системам, функционирующим на основе вычислительной сети. Мало того, можно утверждать, что полезное функционирование РС, не обладающей свойством отказоустойчивости (или обладающей этим свойством в малой степени), попросту невозможно.

Механизмы повышения отказоустойчивоси на уровне компьютера разнообразны, однако все они основаны на избыточности (резервировании) аппаратных средств, программ, информационных массивов, а главное времени, отведенного на решение основных функциональных задач системы. В общем случае отказоустойчивость обеспечивается с помощью следующих механизмов: обнаружение отказа в системе; диагностирование отказавшего устройства; устранение влияния отказавшего устройства (реорганизация

системы); восстановление нормального функционирования системы.

Все эти механизмы являются неотъемлемыми частями отказоустойчивой системы и могут реализовываться аппаратным, программным или смешанным программно-аппаратным способом. В работе рассматриваются указанные механизмы и в качестве базового механизма обеспечения отказоустойчивости рекомендуется откат-восстановление. Более подробно этот механизм описывается в четвертой главе.

В этой же главе служба отказоустойчивости представляется как иерархическая система, состоящая из четырех уровней: виртуального кольца; WATCH - службы; управления фиксацией транзакций; резервирования файлов.

Виртуальное кольцо предназначено для быстрейшего обнаружения изменения состояния узла (т.е. отказа или восстановления) при малом числе пересылаемых сообщений, а также для необходимой простейшей реакции на эти события.

Описывается протокол уровня виртуального кольца, который предназначен для распознавания отказа или восстановления узла. Каждое такое распознавание инициирует работу другого протокола, смысл которого заключается в широковещательном распространении сведений. Этот протокол относится ко второму уровню, т.е. WATCH - службе. Кроме описанного последнего протокола WATCH - служба организует также слежение за состоянием произвольных узлов, заданных прикладным процессом, и "докладывает" о любом изменении состояния узла данному прикладному процессу.

Согласно принципу отказоустойчивости СУТ должна обеспечивать сохранение целостности РБД не только в обычных условиях параллельного выполнения транзакций, но и в условиях отказа одного или нескольких узлов РС. В идеале для прикладных процессов СУТ должна обеспечить "прозрачность" среды по отношению к отказам узлов, т.е. среда должна работать всегда одинаково, отрабатывая ситуацию, связанную с отказом узла, как одну из обычных ситуаций, возникающих в рамках управления параллелизмом.

Рассмотрим вначале случай, когда в РС нет ни службы виртуального кольца, ни WATCH - службы. В этом случае обеспечить отказоустойчивость РС трудно, так как в узлах, как правило, нет достаточной информации о процессах, происходящих в других узлах, поэтому, узнав об отказе одного из узлов, все узлы-участники выполнения транзакции должны провести переговоры для выяснения общей ситуации и определения мер,

необходимых для сохранения целостности РВД.

В работе описывается протокол, известный как протокол двухфазной фиксации транзакций (2PC-протокол), обеспечивающий устойчивость к отказу одного узла. Затем предположение об отсутствии WATCH - службы снимается и строится более устойчивый протокол, также являющийся двухфазным. Описанные протоколы относятся к третьему уровню иерархической системы.

Резервирование данных на уровне различных информационных объектов, например файлов, отношений реляционных БД (или их частей), применяется для достижения двух целей. Во-первых, для приближения информации к пользователю. Тогда узлы будут быстрее реагировать на поступающие сообщения, а значить, будет быстрее функционировать вся РС. Во-вторых, для повышения отказоустойчивости РВД, т.е. при отказе некоторого узла можно использовать копии данных, хранившихся в этом узле, но размещенные в других (функционирующих) узлах. Таким образом, обеспечивается высокая степень доступности данных, минимизируется влияние отказов узлов. Обе эти цели не противоречат друг другу, но тем не менее в ряде случаев требуют различных механизмов работы с резервированными данными.

В работе рассматривается важное и перспективное направление в проблеме отказоустойчивости РС, связанное с предположением о возможности возникновения отказов более сложной природы, чем связанные с неисправностью технических средств сети или ошибками в программах.

В последнее время стремительно развиваются вычислительные системы, предназначенные для управления процессами, протекающими в реальном масштабе времени. Такие системы используются для управления авиарейсами, работой атомных электростанций, телефонных сетей, поточных линий на промышленных предприятиях и т.д. К числу таких систем относится разрабатываемая на междугородной телефонной станции (МТС) Минсвязи Азербайджана автоматизированная система по обработке междугородных телефонных разговоров (автоматика) на базе вычислительного комплекса каналов телефонной станции АРМ-20, персональных ЭВМ (совместимые с IBM PC) и мини-ЭВМ СМ-4. Целью разработки такого специализированного вычислительного комплекса является создание аппаратно-программных средств, предназначенных для эффективной обработки междугородных телефонных разговоров в режиме реального времени.

На данный момент разработана система приема информации из каналов

телефонной станции ARM-20 на базе ПЭВМ типа IBM через стандартный разъем RS-232C. Для повышения надежности системы, круглосуточно принимающей и обрабатывающей оперативную информацию, устанавливаются две ПЭВМ, одна из которых основная, а вторая резервная. Данные ПЭВМ осуществляют прием и контроль поступающей информации, производя отбраковку ошибочных данных с выдачей твердых копий ошибок, распознают и фиксируют разговоры из гостиниц. Далее производится передача проведенной информации на среднюю (третью) ПЭВМ. Последняя производит исключение повторных записей и статистическую обработку разговоров и передает на мини-ЭВМ СМ-4. Поступающая на мини-ЭВМ СМ-4 обработанная информация записывается на магнитные носители с целью дальнейшей обработки.

Разрабатываемый на МТС Минсвязи Азербайджана отказоустойчивый специализированный вычислительный комплекс для обработки междугородных телефонных разговоров позволяет повысить эффективность использования аппаратуры автоматического учета стоимости, оперативно диагностировать и информировать персонал сетевого узла связи о сбоях, происходящих при регистрации и передаче информации на ПЭВМ, автоматического учета стоимости с целью дальнейшей обработки.

Четвертая глава посвящена задачам, решаемым как часть общей проблемы отказоустойчивости, а именно построения контрольных точек - CP (checkpoint) в РС, к которым надлежит вернуть ее после отказа и восстановления технических средств сети, чтобы затем корректно рестартовать и продолжить процесс транзактной обработки.

Предложены и обоснованы принципы использования механизма отката-восстановления как базового программного средства обеспечения отказоустойчивости РС, обладающего достаточной универсальностью, высокой технологичностью использования и позволяющего вернуть систему в непротиворечивое состояние после обнаружения ошибки в работе РС, а затем вновь восстановить ее состояние, имевшееся к моменту обнаружения ошибки, и продолжить нормальное функционирование. После отката система инициирует процесс восстановления. Восстановлением называется приведение системы из непротиворечивого состояния, полученного в результате отката, в непротиворечивое состояние, близкое тому, при котором был обнаружен отказ. Система должна повторно инициировать и выполнять те транзакции, результаты работы которых были аннулированы при откате, и те, которые поступили в систему во время реализации отката и повторного выполнения указанных выше транзакций. Необходимо

отметить, что система возвращается в состояние, близкое тому, при котором был обнаружен отказ, а не точно в то же состояние, ведь повторный процесс с некоторого момента времени (соответствующего моменту возникновения отказа первичного процесса) не повторяет первичный процесс, так как отказа уже не возникает.

В работе выполнен обзор существующих моделей отката-восстановления и даны их сравнительные характеристики. В результате анализа известных методов отката-восстановления отмечены следующие их недостатки:

- возможность возникновения повторных (каскадных) откатов в системах со многими взаимодействующими процессами ("эффект домино"); при этом достаточно общих методов борьбы с указанным явлением не имеется;
- существующие методы определения множества СР для взаимодействующих процессов требуют достаточно жесткой синхронизации последних.

Одним из методов борьбы с "эффектом домино" является формирование (установление) дополнительных СР - ACP (additional checkpoint) перед отработкой сообщений, полученных от другого процесса. Этот подход обеспечивает следующие свойства: процесс никогда не делает двух последовательных откатов без повторной попытки продолжить отработку между ними; дистанция отката мала. Установление ACP требует достаточно высоких затрат машинной памяти и времени. В главе предложен и обоснован эффективный метод, назовем его модифицированным методом ACP, который в ряде случаев дает возможность экономить ресурсы, не поддерживая лишних ACP. Предложенный в работе метод основан на том, что прежде чем установить ACP анализируется ориентированный мультиграф потенциальных откатов. В работе показано, что если в этом графе не существует путь между вершинами одного уровня, то эффект домино невозможен. С учетом этого утверждения СУТ в каждом узле работает согласно следующему основному правилу: если графе не существуют пути между вершинами одного уровня, то ACP не устанавливается, если же такие пути существуют, ACP устанавливается. Информация для формирования в каждом узле мультиграфа рассылается узлами в широковещательном режиме при простановке каждой СР и ACP. В работе предложен еще один протокол простановки автономных СР, позволяющий предотвратить эффект домино.

В предыдущих главах мы практически не интересовались семантикой субтранзакций, выполняемых в РС, однако ряд интересных с теоретической

точки зрения задач связанс простейшим видом транзакций, переносящих из узла-источника в узел-адресат некоторый объект, характеризуемый численным значением, например определенную сумму денег. Здесь мы рассматриваем задачу определения глобального состояния РС, т.е. распределения объектов с их численными значениями между узлами сети (в нашем случае распределением денег по узлам). Неэлементарность задачи состоит в том, что определить глобальное состояние необходимо не останавливая процесс передачи обработки транзакций. Процесс определения глобального состояния можно сравнивать с работой группы фотографов, наблюдающих панорамную динамическую сцену - небо с летающими птицами. Сцена так велика, что ее нельзя охватить одной фотографией. Фотографы должны выполнить несколько моментальных снимков и сложить их вместе для получения общей картины. Фотографы не должны нарушать сам фотографируемый процесс: например, остановить на некоторое время полет птиц. Кроме того, фотографы не имеют службы общего времени и не могут обеспечить спуск затворов фотоаппаратов точно в один и тот же момент времени.

По аналогии с процессом фотографирования такой динамической панорамы процесс определения глобального состояния обычно называют "моментальным снимком" - SS (snapshot state) РС. Задача протокола "моментального снимка" - определить некоторое динамическое глобальное состояние. При этом тонкость заключается в том, что это "некоторое" глобальное состояние могло не существовать ни в один реальный момент времени, т.е. эта абстракция. Однако эта абстракция достаточно полезна, т.к. можно доказать, что полученное в результате выполнения протокола динамическое глобальное состояние могло существовать и на основании этого можно построить практические механизмы его использования.

В работе предложено обобщение протокола "моментального снимка", фиксирующего глобальное состояние сети на основе "перекрашивания узлов". Принимается, что каждый узел и каждое сообщение могут быть окрашены либо в белый, либо в красный цвет. Важно то, что сообщения посылаемые белым узлом - белые, и наоборот. В начальный момент все узлы и сообщения имеют одинаковую окраску, например, белую. После инициации некоторым узлом протокола моментального снимка начинается "перекрашивание" всех узлов сети в красный цвет. Отметим, что цвет узла и его сообщений не несет смысловой нагрузки. Цвет используется в данном случае только для снятия моментального снимка состояния узл

как своеобразная метка, разграничивающая периоды записи состояния РС в процессе ее безостановочной работы. Протокол обладает следующими свойствами, обеспечивающими в совокупности его корректность.

1. Очередная перекраска заканчивается за конечное время.

2. Если в процессе "перекраски" один или несколько узлов по своей инициативе возбудит одноименный процесс, то это не нарушит результата протокола.

3. Если в очередном процессе "перекраски" один или несколько узлов по своей инициативе возбудит противоположный процесс, то это не воспрепятствует завершению очередного процесса.

В работе на основе В-протокола приведена формализованная запись протокола моментального снимка.

Идея определения глобального состояния РС лежит в основе решения проблемы формирования глобальных СР - GCP (global checkpoint), которые являются обобщением понятия СР вычислительного процесса на распределенную вычислительную систему.

GCP служит не только для рестарта РС при отказах оборудования, но и является основой решения некоторых специфических прикладных задач в РС, например для ревизии счетов распределенной банковской системы, переписи населения и др.

Процесс формирования GCP инициируется одним из видов транзакций. Для того чтобы транзакция формирования GCP дала значимый результат, отдельные этапы формирования GCP должны быть четко (скоординированы) с этапами выполнения других (обычных) транзакций. Например, для банковской системы в результате произвольного чередования операций формирования GCP и операций по переводу денег некоторые переводы могут быть полностью пропущены, а некоторые переведенные деньги могут быть сосчитаны дважды. Таким образом, для суммы всех счетов будет получено неверное значение.

Предлагается модель систем управления распределенными транзакциями и на основе этой модели анализируются существующие различные подходы к формированию GCP. Все они отличаются друг от друга реализацией, продолжительностью простоя глобальной СУТ для формирования GCP, сложностью механизма отката-восстановления и т. д.

Далее в главе приводятся протоколы формирования GCP. Формируемые GCP нумеруются по порядку: GCP(r), где r=1,2,.... Все транзакции и порожденные ими субтранзакции принадлежат одной и только одной GCP, что и фиксируется включением численного аргумента, равного номеру

соответствующей GCP(r). Этот номер будем называть CPS(checkpoit stamp) и обозначать CPS - r.

Согласно определению глобального состояния необходимо во всех локальных БД зафиксировать состояние, соответствующее формированию GCP(r), после успешного или аварийного завершения всех транзакций с меткой CPS- r-1. В общем случае в различных узлах DM такие состояния достигаются в различные моменты времени, однако в данном случае можно применить принцип "каждый отвечает за себя", т.е. если некоторый DM закончил отработку субтранзакций с меткой CPS - r-1, то он может фиксировать состояние, не интересуясь тем, что в этот момент происходит в других DM, и продолжить работу, соответствующую периоду [GCP(r),GCP(r+1)]. Обоснованием этого принципа является правила службы параллельного управления - CC (concurrency control), например, протокол двухфазной фиксации. Согласно этим правилам считаем, что если в некотором DM субтранзакции завершены, то ни при каких условиях, вплоть до отказов других DM , решение о виде ее завершения (успешном или аварийном) не меняется.

Предложенные в работе протоколы отличаются друг от друга следующими правилами. В протоколе P1 ни одна субтранзакция с мекой CPS - r не будет выполнена до завершения всех субтранзакций с меткой CPS - r-1. В отличие от этого в протоколе P2 субтранзакции с меткой CPS - r, не конфликтующие с субтранзакциями с меткой CPS - r-1, начнут выполняться через сравнительно короткий интервал времени после получения в данном узле DM сообщения о необходимости формирования GCP(r). Протокол P3 отличается тем, что в нем не предусмотрены откаты субтранзакций, а также устойчивостью к отказам DM.

Предложен протокол формирования GCP в PC с древовидной топологией. Древовидная топология сети часто встречается в иерархических региональных и/или отраслевых системах. Инициатором формирования GCP является центральный (высший по иерархии) узел - корень дерева. Запрос на формирование GCP он рассылает только множеству непосредственно подчиненных ему по иерархии узлов, те, в свою очередь,-множеству подчиненным непосредственно им и т. д. Данный протокол основан на том, что новые CP строятся только в тех узлах, между которыми был осуществлен обмен сообщениями ,приведшими к изменению БД.

Утверждение. Не участвующий в обмене сообщениями процесс не обязан строить CP при поступлении централизованной команды.

Предложенный протокол позволяет минимизировать количество

сообщений, обменивающихся между процессами и экономить память от сохранения лишних СР.

Пятая глава посвящена описанию блоков СУТ в АСОИ Азербайджана. Технической базой этой системы является трехузловая экспериментальная сеть СМ ЭВМ. Основная цель создания сети - практическая обработка научных исследований в области теории и практики создания сетей и сетевых протоколов, экспериментальная проверка системного программного обеспечения республиканской сети вычислительных центров и передачи данных, реализация распределенной информационно-справочной системы АН Азербайджана.

АСОИ АН состоит из следующих компонентов: сеть передачи данных, прикладное программное обеспечение (ПО) и администратор системы. СПД реализована на базе локальной вычислительной сети Алиса+, включающей в себя ПЭВМ IBM PC/AT, СМ1420 и СМ1403, подсистему информационного обслуживания зала заседаний на базе электронного табло коллективного пользования, управляемого ПЭВМ ЕС1841, а также линии связи внутренней и городской АТС с соответствующими коммуникационными оборудованиями.

Экспериментальная сеть СМ ЭВМ разработана на основе эталонной семиуровневой модели открытых систем и удовлетворяет основным требованиям сетевой архитекуры СМ ЭВМ и DNA. Передача информации по сети осуществляется пакетом программ, который обеспечивает межсетевое взаимодействие как с сетями типа DECNET, так и с сетями, разработанными на основе рекомендаций Х.25 МККТТ.

Верхний уровень иерархии ПО представляет сетевой администратор базы данных (САБД), основные функции которого заключаются в управлении транзакциями в среде РБД, поддержке непротиворечивости РБД, синхронизации актуализации файлов РБД и формировании заданий на передачу информации по сети.

САБД разработан для функционирования в экспериментальной сети СМ ЭВМ и обеспечивает пользователю доступ к БД, распределенной физически в различных узлах сети. Для обеспечения функций САБД в каждом узле хранится системная справочная информация, включающая описания логической структуры РБД, схему размещения файлов по узлам сети, а также схемы размещения заданий и абонентов сети. Процедуры САБД обеспечивают следующие функции управления БД:

- ведение таблиц сетевых данных (системных справочников) и пересылка во все узлы сети;
- управление доступом к РБД;

- формирование запросов на передачу данных по сети ЭВМ;
- поддержка непротиворечивоси РБД;
- ведение процесса актуализации РБД;
- учрежденческая электронная почта;
- ведение единого сетевого времени;
- ведение статистики функционирования РБД.

СУТ представляет собой независимую от конкретно применяемой СУБД программу, находящуюся в оперативной памяти на всем протяжении работы АСОИ АН. СУТ выполняется во всех основных вычислительных узлах экспериментальной сети как основная часть САВД, обеспечивающая взаимодействие прикладных процессов пользователя с РБД. СУТ относится к верхнему уровню семиуровневой архитектуры МОС. Главной задачей СУТ является обеспечение непротиворечивости РБД. Воспринимая обширный поток входной информации от конечных пользователей такая система должна при любом отказе технических средств и/или ошибочных действиях операторов сохранить целостность РБД, правильно прекратить выполнение начатых транзакций и привести систему к такому состоянию, с которого можно продолжить нормальную работу.

Транзакция порождается в некотором узле ТМ и выполняется в одном или нескольких других узлах DM . Считаем, что любой ТМ в любой момент времени может связаться (т.е. послать сообщение) через сеть передачи данных с любым вычислительным узлом DM. СПД предполагается абсолютно надежным устройством, доставляющим сообщение из конца в конец за время, ограниченное конечным интервалом Т. В любой момент времени в РС существует не более одного координатора. Роль координатора может выполнять любой из узлов DM. Если координатор отказывает, то все остальные DM завершают начатые транзакции в соответствии со своими протоколами, гарантирующими атомарность транзакции и целостность всей РБД и приступают к выборам нового координатора. Пока не будет выбран новый координатор, ни одна транзакция не может быть начата.

Прикладные процессы в узлах ТМ формируют запросы на выполнение транзакций. Узел ТМ связывается с координатором для получения разрешения на использование РБД. Узел ТМ, получив разрешение от координатора, рассылает узлам DM поручения по обработке РБД и управлению сообщениями в соответствии с тем или иным протоколов СУТ. Узлы DM, руководимые ТМ, выполняют транзакцию, реализуя необходимые изменения РБД. Координатор создает и ведет очередь запросов на ресурсы

и определяет, может ли та или иная транзакция в настоящий момент выполняться.

Выполнение любой обработки информации, хранящейся в РБД, производится через интерфейс "СУТ-ПП". Для выполнения атомарного действия над частью РБД используется интерфейс "СУТ-ПП". Для стыковки СУТ с СПД служит администратор сети, включающий модуль-инерфейс с САБД и ПО управления СПД.

Интерфейс "СУТ-ПП" используется в двух случаях: а) для перевода пользовательского запроса на выполнение транзакции в последовательность атомарных действий с целью последующего распределения выполнения транзакции по узлам DM; б) для выполнения атомарного действия над частью РБД в любом узле DM. Все переговоры с СУБД в случае а) ведет координатор, в случае б) - любой из узлов DM, решивший выполнять транзакцию, должен сам связаться с СУБД. Если схема РБД известна в любом узле TM, то необходимость случая а) отпадает; запрос вырождается в список узлов DM, необходимых для осуществления работы данной транзакции.

Для случая б) (с точки зрения DM):

1. Получив сообщение от TM (содержащее описание атомарного действия) узел DM изучает возможность частичного выполнения атомарного действия. В случае отказа от выполнения узел DM шлет TM ответ "no" и снова переходит в состояние ожидание сообщений от TM.

2. Если узел DM решает выполнять транзакцию, он связывается с СУБД, передавая ей запрос на выполнение. СУБД производит необходимые действия и записывает результат в рабочую область узла DM, либо сообщает об ошибке.

СУБД доступна через монитор MA(РБД). Любой из узлов связывается с другими узлами через СПД посредством сетевого монитора MA(СПД), реализующего уровни I-VI.

СУТ состоит из следующих основных блоков: блок запуска транзакции; блок выполнения транзакции; блок определения состояний; блок общего времени; блок обеспечения живучести; блок журнализации; коммуникационно-управляющий блок; блок выбора координатора; блок фиксации результатов.

В заключении изложены основные научные и практические результаты, полученные в диссертационной работе.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Основные результаты, полученные в диссертационной работе таковы:

1. Сформулирована в общем виде задача синхронизации процессов как построение функции упорядочения событий в распределенной системе в соответствии с некоторыми содержательными критериями, специфическими для данной задачи. Рассмотрены основные утверждения, которые должны быть доказаны при решении практически всех частных задач синхронизации.

2. Предложены общие принципы построения распределенного плана выполнения транзакций над РБД, удовлетворяющего свойству сериализуемости. Показано, что обеспечение этого свойства является условием для построения функции упорядочения событий в РС. Содержательным критерием при этом является условие ненарушения целостности РБД.

3. Описан базовый протокол взаимодействия процессов в РС, представляющий собой основу для построения полных протоколов различной функциональной направленности. Построен протокол, позволяющий заранее предупредить возникновение несериализуемого распределенного плана.

4. Разработан и обоснован ряд протоколов в РС, решающих задачу сериализуемости, основанных на временных метках Лэмпорта и проверки на ацикличность графа, отражающего упорядоченность транзакций во времени и их информационную зависимость.

5. Предложена единая иерархическая структура механизмов обеспечения отказоустойчивости. Построены протоколы фиксации транзакций в условиях параллельного их выполнения, позволяющие сохранить целостность РБД как при исправной сети, так и при отказе отдельных ее узлов.

6. Предложены и обоснованы принципы использования механизма отката-восстановления как базового программного средства обеспечения отказоустойчивости РС, обладающего достаточной универсальностью высокой технологичностью использования и позволяющего вернуть систему в непротиворечивое состояние после обнаружение ошибки в работе РС, а затем вновь восстановить ее состояние, имевшееся к моменту возникновения ошибки, и продолжить нормальное функционирование. Предложена формальная модель РС и проведено исследование механизма отката-восстановления с применением аппарата контрольных точек.

7. Проведен сравнительный анализ существующих методов отката-восстановления и на основе вышеуказанной модели предложены методы

отката-восстановления взаимодействующих процессов, позволяющих предотвратить "эффект домино".

8. Предложена формальная модель СУТ, на основе которой исследовано глобальное состояние системы и проведен сравнительный анализ существующих алгоритмов обеспечения непротиворечивости РБД.

9. Обобщен протокол "моментального снимка", фиксирующий глобальное состояние сети по Лэмпорту на основе "перекрашивания" узлов.

10. Разработаны протоколы формирования глобальных контрольных точек РБД, к которым надлежит вернуть систему после отказа и восстановления технических средств сети, чтобы затем корректно рестартовать и продолжить процесс транзактной обработки. Эти протоколы сводят к минимуму число откатываемых транзакций и уменьшают простой РС при формировании глобальных контрольных точек и, следовательно, повышают производительность СУТ.

11. Предложен рациональный протокол определения глобального состояния сети и глобальной контрольной точки. Протокол отличается тем, что в нем не предусмотрены откаты субтранзакций, а также наличием логических связей с протоколом двухфазной фиксации транзакций.

12. Разработанные методы и протоколы реализованы в виде блоков системы управления транзакциями в распределенной АСОИ АН Азербайджана.

13. Разработанная специализированная вычислительная система для эффективной обработки междугородных телефонных разговоров в режиме реального времени внедрена на МТС Минсвязи Азербайджана.

Основные результаты диссертации опубликованы в следующих работах

1. Алиев А.А. Эффект домино в распределенных системах //Вычислительные сети коммутации пакетов: Тез. докл. У Всесоюз. конф. - Рига, 1987. - Ч.2 - с.243-244.

2. Алиев А.А. Координация взаимодействующих процессов в распределенных системах реального времени //Методы анализа надежности программного обеспечения вычислительных систем реального времени на основе моделей нечеткой логики и качественных описаний: Тез. докл. I Всесоюз. науч.-техн. конф. - Киев, 1987. - с.121.

3. Алиев А.А., Никитин А.И. Откат в распределенных системах реального времени //Математическое, программное и техническое обеспечение вычислительных систем: Сб. науч. тр. - Киев: Ин-т

кибернетики им. В. М. Глушкова АН Украины, 1987. - с. 28-33.

4. Алиев А. А. Контрольные точки и откат-восстановление в распределенных системах //УСиМ. - 1988. - N4. - с. 30-36.

5. Никитин А. И., Алиев А. А., Гостилова С. В. Глобальное непротиворечивое состояние распределенных баз данных. - Киев, 1988. 21 с. - (Препр. /АН УССР. Ин-т кибернетики им. В. М. Глушкова: 88-48).

6. Алиев А. А. Механизмы отказоустойчивости распределенных систем //Конф. молодых ученых и специалистов Института кибернетики им. В. М. Глушкова АН УССР, 1988. - с. 8-11. - Деп. в ВИНИТИ 21.02.89, N1120-В89.

7. Алиев А. А. Глобальные контрольные точки в распределенных базах данных //Живучесть и конфигурация информационно-вычислительных систем: Тез. докл. II Всесоюз. науч.-техн. конф. - Алушта, 1988. - с. 101.

8. Аббасов А. М., Алиев А. А., Дадашев В. Э. Распределенная база данных автоматизированной системы обработки информации АН АзССР //Вычислительные сети коммутации пакетов: Тез. докл. VI Всесоюз. конф. - Рига, 1989. - с. 333-337.

9. Алиев А. А., Иманова Н. Ф. Определение глобального состояния распределенных систем //Тез. докл. Республ. науч.-практ. конф. - Баку, 1990. - с. 37.

10. Алиев А. А., Аскеров В. А., Асланова Н. М., Иманова Н. Ф. Отказоустойчивый специализированный вычислительный комплекс для обработки междугородных телефонных разговоров (автоматика) //Тез. докл. I Всесоюз. науч.-практ. семин. - Харьков, 1990. - с. 123-127.

11. Алиев А. А. Глобальное состояние распределенных систем. - Киев, 1993. - 25 с. - (Препр. /АН Украины. Ин-т кибернетики им. В. М. Глушкова; 93-41).

12. Алиев А. А., Гостилова С. В., Никитин А. И., Попович В. Л. Алгоритмы обеспечения сериализуемости распределенного плана в системах управления транзакциями //УСиМ. - 1995. - N1-2. - с. 37-39.

13. Никитин А. И., Алиев А. А. Синхронизация прикладных процессов в распределенных системах обработки данных. - Киев: Аграрная наука, 1995. - 128 с.

14. Попович В. Л., Алиев А. А. Оптимистический подход к обеспечению сериализуемости параллельных транзакций в распределенных системах /Ин-т проблем мат. машин и систем НАН Украины. - Киев, 1995. - 5 с. - Деп. в ГНТБ Украины от 10.05.95, N 1081.

15. Алиев А. А. Сериализуемость распределенного плана выполнения

транзакций /Ин-т проблем мат. машин и систем НАН Украины. - Киев, 1995. - 26 С. - Деп. в ГНТБ Украины от 26.05.95, N 1264.

16. Алиев А.А., Гостилова С.В., Иманова Н.Ф. Методы управления параллелизмом транзакций в распределенных системах //Представление знаний в информационных технологиях: Сб. науч. тр. - Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины, 1995. - с. 150-156.

17. Алиев А.А., Попович В.Л. Два метода управления параллелизмом транзакций в распределенных системах //Сб. науч. тр. - Киев: Ин-т кибернетики им. В.М. Глушкова НАН Украины. (в печати).

18. Никитин А.И., Алиев А.А., Попович В.Л., Гостилова С.В. Прямой метод обеспечения сериализуемости распределенного плана выполнения в системах управления транзакциями //Математические машины и системы. (в печати).

А Н О Т А Ц І Я

Аліев А. А. Протоколи синхронізації прикладних процесів в розподілених системах обробки інформації. Дисертація на здобуття наукового ступеню доктора технічних наук зі спеціальності 05.13.04 - автоматизовані системи управління і системи обробки інформації, Київ,1996

Дисертаційна робота присвячена питанням розвитку теорії взаємодії і синхронізації прикладних процесів в розподілених системах обробки інформації. В роботі визначається в загальному вигляді задача синхронізації як побудова функції впорядкування подій в системі. Розглянуті основні твердження, які повинні бути доведені при вирішенні практично всіх задач синхронізації. Вводиться поняття розподіленого плану виконання субтранзакцій, а також властивості серіалізуемості цього плану. Приводиться та аналізується ряд протоколів,що вирішують задачу серіалізуємості. Вводяться основні поняття відмовостійкості і механізми досягнення останньої. Запропонована єдина ієрархічна структура механізмів забезпечення відмовостійкості розподілених систем та побудовані протоколи фіксації транзакцій в умовах їх паралельного виконання. Розглядається проблема визначення глобального стану розподіленої системи при виконанні паралельних транзакцій і при можливому виході з ладу одного чи більше вузлів системи. Розроблені протоколи,що дозволяють не знижуючи продуктивності системи, формувати глобальні контрольні точки, які відображують глобальний цілісній стан розподіленої бази даних. Описані рівноманітні блоки системи управління транзакціями, які забезпечують цілісність розподіленої бази даних автоматизованої системи обробки інформації АН Азербайджану.

Ключові слова: синхронізація,розподілена система,відмовостійкість, цілісність, глобальний стан, контрольна точка, транзакція.

ANNOTATION

Aliyev A. A. Synchronization protocols of application processes in distributed systems of information processing. Dissertation for academic degree of Doctor of technical sciences on speciality 05.13.04. - automation control systems and systems of information processing, Kiev, 1996.

The dissertation is dedicated to questions of theory development of interaction and synchronization of application processes in distributed systems of information processing. Task of synchronization as construction of regulation function of events in the system is defined in the general form in this work. Main affirmations which must be proved under decision of almost all synchronization tasks have been considered work. Conception of distributed plan od sub-transactions fulfilment, and also characteristics of serializability of this plan are introduced. Number of protocols solving serializability task is considered and analyzed. Main conceptions of fault-tolerance and methods to achieve it are introduced. Integrated hierarchical structure of methods to provide fault-tolerance of distributed systems has been offered and protocols of transactions commit in conditions of their parallel fulfilment have been built. Problem of determination of globa of distributed system under parallel transactions fulfilment and possible disrepair of one or more system notes is considered. Protocols which allow to form global checkpoints reflecting global consistency of distributed data base have been designed. Various blocks of transactions control system providing consistency of distributed data base of automation information process ng system of Academy of Sciences of Azerbaijan have been described.

Key-words: synchronization, distributed system, fault-tolerance, consistency, global state, checkpoint, transaction.

ANNOTATION

Aliyev A.A. Synchronization protocols of application processes in distributed systems of information processing. Dissertation for academic degree of Doctor of technical sciences on speciality 05.13.04. - automation control systems and systems of information processing, Kiev, 1996.

The dissertation is dedicated to questions of theory development of interaction and synchronization of application processes in distributed systems of information processing. Task of synchronization as coordination of [illegible] of events in the system is defined in the general form [illegible] of relations which must be [illegible] synchronization tasks have been considered. [illegible] distributed plan od sub-transactions fulfilment, and also characteristics of serializability of this plan are introduced. Number of protocols solving serializability task is considered and analyzed. Main conceptions of fault-tolerance and methods to achieve it are introduced. Integrated hierarchical structure of methods to provide fault-tolerance of distributed systems has been offered and protocols of transactions commit in conditions of their parallel fulfilment have been built. Problem of determination of globe of distributed system under parallel transactions fulfilment and possible disrepair of one or more system notes is considered. Protocols which allow to form global checkpoints reflecting global consistency of distributed data base have been designed. Various blocks of transactions control system providing consistency of distributed data base of automation information processing system of Academy of Sciences of Azerbaijan have been described.

Key-words: synchronization, distributed system, fault-tolerance, consistency, global state, checkpoint, transaction.