

На правах рукописи



ЗУДИН АНДРЕЙ БОРИСОВИЧ

**ПРИМЕНЕНИЕ АСПЕКТНО-ОРИЕНТИРОВАННОГО
ПОДХОДА К СОЗДАНИЮ ОРТОГОНАЛЬНЫХ
ДОЛГОВРЕМЕННЫХ ОБЪЕКТНЫХ СИСТЕМ В
АРХИТЕКТУРЕ С ВЕБ-КЛИЕНТОМ**

**Специальность 05.13.11 - Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей**

АВТОРЕФЕРАТ

**диссертации на соискание ученой степени
кандидата технических наук**

20 ЯНВ 2011

Москва 2010 г.

**Работа выполнена на кафедре «Вычислительные системы и сети»
Московского государственного института электроники и математики**

Научный руководитель:

Саксонов Евгений Александрович
доктор технических наук, профессор

Официальные оппоненты:

Каширин Игорь Юрьевич
доктор технических наук, профессор

Авдошин Сергей Михайлович
кандидат технических наук, доцент

Ведущая организация: Институт системного анализа РАН

Защита состоится «25» января 2010 г. в 14 часов на заседании диссертационного совета Д 212.133.01 при Московском государственном институте электроники и математики (МИЭМ) по адресу 109028, Москва, Б. Трехсвятительский пер., дом 3.

С диссертацией можно ознакомиться в библиотеке МИЭМ.

Автореферат разослан «22» декабря 2010 г.

Ученый секретарь
диссертационного совета,
кандидат технических наук,
доцент



Бузников С.Е.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Задача управления данными, у которых время жизни может превышать время жизни программной системы, которая оперирует ими, является важной и в теоретическом и в практическом плане. Сопряжение кратковременных вычислений с долговременными данными поднимает ряд фундаментальных вопросов типа и тождественности. Также возникает целый ряд важных инженерных вопросов в связи с необходимостью эффективно и согласовано перемещать данные между временными и надежными хранилищами.

Для работы с подобными данными создаются системы долговременного программирования. Одной из важных задач долговременного программирования является создание программных средств, которые оперируют долговечными, конкурентно доступными и потенциально большими объемами информации. Такие программные средства принято называть долговременными (ДПС) (Persistent Application Systems (PAS)). Типичными представителями ДПС являются, например, CAD/CAM системы, системы автоматизации офисов, CASE средства, среды разработки программных средств, интегрированные системы управления медицинскими учреждениями, большие научные базы данных и программы их анализа, географические информационные системы, системы моделирования окружающей среды, объектно-ориентированные базы данных и системы моделирования процессов.

Среди известных подходов наибольший интерес представляют ортогональные системы долговременного программирования (ОСДП) и их развитие – объектные ОСДП.

ОСДП предназначены улучшить системную архитектуру, упростить поддержку и увеличить производительность ДПС. Абстрагирование от времени жизни данных позволяет создавать и манипулировать объектами независимо от их типа и посредством этого совместить программную модель данных и модель данных СУБД. В результате увеличивается продуктивность разработчиков ДПС систем.

Объектные ОСДП (ООСДП), появлению которых во многом способствовало развитие Интернет, ставят перед своими разработчиками множество различных вопросов. Особенностью работы с долговременными объектами является то, что она затрагивает такие аспекты, которые не отражены в

универсальных алгоритмических языках программирования: представление объектов во внешней памяти, параллельный доступ, эффективность и т.д. Эти аспекты всегда будут влиять на ортогональность программного кода работающего с временными или с долговременными объектами.

Аспектно-ориентированный подход позволяет минимизировать отличия программного кода для работы с разными объектами и, тем самым, упростить повторное использование кода.

Представленная в работе ООСДП на основе тонкого клиента, являясь эффективной реализацией ортогональной стабильности объектов, является так же и средой разработки и выполнения для информационных Интернет систем, где вся разработка может вестись на основе клиентского языка сценариев, не требуя изменений на других уровнях многозвенной архитектуры.

Опираясь на известные данные о разработке программных систем, можно сделать вывод, что ДПС приобретают все большую социальную и экономическую значимость. Они потенциально обладают длительным временем жизни, средствами поддержки одновременного доступа со стороны пользователей и состоят из большого объема данных и программного кода. В связи с этим актуальными являются исследования, направленные на разработку принципов и подходов к построению персистентных систем.

Целью работы является исследование свойств систем долговременного программирования и создание прототипа ортогональной объектной системы долговременного программирования на основе аспектно-ориентированного подхода, позволяющего эффективно организовать хранилища объектов для информационных систем с тонким клиентом, функционирующего на последнем уровне многозвенной архитектуры.

Для достижения поставленной цели в работе сформулированы и решены следующие задачи:

- исследованы возможности и особенности существующих универсальных ООСДП и на базе этого анализа, с учетом требований, предъявляемых тонкими клиентами, построена метамодель для создания, хранения и управления доступом к объектам в хранилище;

- исследована возможность реализации ООСДП, построенной на основе этой метамодели, и возможность реализации в рамках данной метамодели различных методов управления доступом к объектам, удовлетворяющих требованиям различных классов приложений.

- определены базовые метаклассы, реализующие наиболее распространенные методы управления объектами и позволяющие легко определять собственные стратегии, а также эффективные алгоритмы для объектного хранилища, обеспечивающие высокую степень масштабируемости и надежности, возможность распараллеливания процессов выполнения объектных запросов;

- разработаны модели параллельной обработки запросов, позволяющие оценить значения характеристик процесса исполнения запросов;

- показаны практические возможности выполнения соответствия требованиям парадигмы повторного использования, обеспечения всестороннего контроля при работе с долговременными объектами, поддержки распределенной работы для обеспечения работы произвольного числа клиентов, расположенных на различных узлах сети.

На защиту выносятся:

1. Результаты анализа систем долговременного хранения, выделяющие основные свойства систем, взаимосвязи между программным обеспечением, свойствами объектов и методами хранения.
2. Методы организации параллельной обработки и требования к языкам программирования для работы с долговременными объектами и их реализация на основе аспектно-ориентированного подхода.
3. Математические модели параллельной обработки запросов в системах долговременного программирования.
4. Методы программной реализации прототипа ортогональной объектной системы долговременного программирования.

Научная новизна полученных результатов состоит в выявлении связей и установлении закономерностей при создании и реализации систем долговременного программирования на основе аспектно-ориентированного подхода, разработке концепции прототипа ООСДП для работы Web-клиентов. Разработке математических моделей для расчета характеристик процесса параллельной обработки запросов.

Практическая ценность результатов диссертации заключается в разработке рекомендаций и конкретных решений для создания компонент программного обеспечения системы долговременного программирования.

Достоверность и обоснованность результатов диссертации обусловлены их согласованностью с известными данными, опубликованными в

работах отечественных и зарубежных авторов, и подтверждаются данными об успешном применении при разработке конкретного программного обеспечения различного назначения.

Методы исследований

При решении поставленных в диссертации задач применялись методы теории систем, теории очередей, системного программирования, объектного программирования.

Реализация и внедрение результатов

Результаты, полученные при выполнении диссертационной работы, использовались в ООО «X-RING» при создании специализированного программного обеспечения для МВД России, при создании компонентов системы долговременного программирования в Интернет системе для Торгово-промышленной палаты России.

Апробация работы

Основные результаты диссертации докладывались и обсуждались на отечественных и зарубежных научно-технических конференциях: Международная конференция «Ломоносов-2003», (Москва, 2003), Научная конференция «Ломоносовские Чтения», Секция Вычислительной Математики и Кибернетики, (Москва, НИВЦ, 2003), XLVI Всероссийская конференция по проблемам математики, информатики, физики и химии. (Москва, РУДН, 2010), 2-я Международная научно-практическая конференция. Современные информационные компьютерные технологии – мсIT – 2010, (Беларусь, Гродно, 2010).

Публикации

Результаты диссертации отражены в 9 опубликованных печатных работах.

Структура и объем диссертации

Диссертация состоит из введения, четырех глав, списка литературы и приложения. Объем диссертации 137 страниц.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность направления исследований, проводимых в диссертационной работе, сформулированы цели и задачи исследований, показаны научная новизна и практическая значимость полученных результатов.

В первой главе диссертации показано, что в связи с появлением новых аппаратных технологий с абсолютно другими компромиссами скорости вычислений, количества ресурсов и стоимости, а так же в ответ на повсеместное использование распределенных вычислений, приходит время разработки новых моделей вычислений и долговременные системы – одна из таких моделей.

Показаны принципиальные отличия между ДПС и известными подсистемами поддержки.

Исследованы свойства и отличительные особенности ортогональных систем долговременного программирования (ОСПД).

Показано, что ОСДП, как средства создания ДПС, отличаются от СУБД и языков программирования СУБД тем, что дают разработчику возможность пользоваться долговременными данными независимо от их назначения. Эта независимость становится возможной при следовании следующим трем принципам: независимости, ортогональности типа, определения времени жизни. Разработчики языков программирования для ОСДП сталкиваются с задачами унификации системы типов ЯП и моделей данных СУБД, построения механизмов именования сохраненных объектов и задачей дальнейшего развития систем.

Обосновано применение аспектно-ориентированного подхода, позволяющего вынести код, отвечающий за реализацию аспектов, на уровень метаклассов, что позволяет минимизировать отличия программного кода для работы с разными объектами и, тем самым, упростить повторное использование кода.

Исследования показали, что популярной архитектурой распределенных вычислений на сегодняшний день являются многозвенные архитектуры с тонким клиентом, например, Интернет браузерами, которые не требуют изменений с развитием системы, опираются на международные стандарты и не предъявляют больших системных требований. Исследованы возможности и особенности существующих универсальных ООСДП и на базе этого анализа с учетом требований, предъявляемыми тонкими клиентами, строится метамодель для создания, хранения и управления доступом к объектам в хранилище.

Показано, что реализацию ООСДП, построенной на основе метамодели, целесообразно проводить в соответствии со следующими требованиями: возможность реализации в рамках метамодели различных методов управления доступом, использование базовых метаклассов, реализующих наиболее распространенные методы управления объектами.

Отмечается, что в долговременных системах существует три правила проектирования, которые имеют свои корни: скупость концепций, ортогональность концепций, полнота вычислительного окружения. Выделены принципы проектирования долговременных систем: принцип независимости, принцип ортогональности к типу, принцип определения времени жизни.

Изучены особенности объектные ОДС, связанные с возможностью унифицированной работы с объектами любых типов, позволяющей значениям любого типа иметь произвольное время жизни. Основной особенностью современных ООСДП является предоставление разработчику приложения всех средств современных объектно-ориентированных языков программирования для работы с долговременными объектами. При этом, с самого начала развития ООСДП, наблюдалась тенденция использования хорошо известных ЯП, таких как C++, Smalltalk и, в последнее время, Java.

Для того, чтобы ООСДП могли быть использованы для как можно более широкого класса приложений необходимо сделать их расширяемыми, т.е. предоставить разработчику возможность определять собственные методы управления доступом к хранимым объектам, отвечающие потребностям конкретного приложения. Проведен обзор проблемы и выбор путей решения.

Показано, что основная проблема, с которой встречаются создатели системы типов для долговременных систем, заключается в необходимости предоставить выразительно мощную модель данных, обладающую совершенной системой защиты, и при этом описываемую в как можно более статической форме. Разработаны последовательные шаги для достижения этих двух противоречащих друг другу целей: Шаг 1. Конструкторы типов. Шаг 2. Скрытие информации о типе при помощи типовых операторов, связанных квантором существования (абстрактных типов). Шаг 3. Использование полиморфизма.

При этом наиболее применимыми являются параметрический полиморфизм и полиморфизм включения, знакомый по его применению в большинстве современных объектно-ориентированных языках.

При параметрическом полиморфизме можно определять процедуры и функции, которые применимы к различным типам параметров. Например, функция, вычисляющая количество элементов списка. Поскольку такая функция не будет использовать сами значения элементов списка, она будет универсальной для любого списка. Тогда эту функцию можно записать в следующем виде

function Count : $\forall T \Rightarrow (\text{List}[T] \rightarrow \text{int})$. Преимуществом построения абстрактных функций является то, что такую функцию можно поместить в хранилище и использовать по мере надобности для вычисления размера любых списков.

Показана необходимость динамической проверки соответствия типов в долговременных системах и предложены способы ее реализации. С ростом количества типизированных файлов в системе, программа будет все чаще работать не с одним, а с несколькими файлами и появится необходимость межфайловых ссылок. Это является одним из главных отличий между хранилищем типизированных файлов и объектов. Файлы являются автономными хранилищами данных, и сами по себе не поддерживают ссылки на другие файлы, в то время как объекты обычно сильно связаны друг с другом.

Исследованы механизмы связывания. Связывание используется для достижения следующих результатов: создание новых объектов (и их помещение в долговременное хранилище), повторное использование существующих в хранилище объектов, создание новых комбинаций из существующих или/и новых объектов, пошаговое создания объектов. Показано, что механизмам связывания приходится учитывать такие факторы, как, например, указание на то, является ли значение константой или переменной; время связывания; время проверки соответствия типов, контекст.

Исследованы задачи обеспечения параллельных вычислений в долговременных системах. Задача эффективного и согласованного переноса данных из долговременного хранилища в оперативную память и обратно усложняется необходимостью параллельных вычислений. Ее можно поделить на две различных подзадачи: задача поддержки одновременного переноса данных в хранилище и вычислений с этими данными; задача поддержки параллельных зависимых вычислений оперирующими кратковременными данными.

Показано, что принцип независимости ортогональных долговременных систем и невозможность создавать простейшую (ACID) транзакцию в контексте другой транзакции приводят к тому, что минимальной транзакцией будет являться создание нового процесса средствами операционной системы. Это существенно ограничивает возможности параллельных вычислений и тем самым вступает в противоречие с другим ключевым принципом ортогональных долговременных систем: принципом вычислительной полноты.

Во второй главе исследованы общие принципы аспектно-ориентированного программирования и возможности их применения при создании систем долговременного программирования. Известно, что основным способом построения программных систем на сегодня остается разбиение на составные компоненты, такие как блоки, процедуры, модули, объекты, звенья (клиенты и серверы – компоненты). Все эти понятия, как правило, напрямую соответствуют блокам исполняемого кода программы. Однако, возрастающая сложность программного обеспечения приводит к тому, что современная сложная система должна предоставлять решения сразу для нескольких классов задач, таких как управление распределенными данными, синхронизация совместного доступа, защита информации, обеспечение надежности и т.д. При попытке воспользоваться при проектировании такой системы стандартным подходом разбиения на модули разработчик сталкивается с рядом вопросов: по какому принципу производить разбиение на модули, какой модуль отвечает за восстановление после сбоя, какой модуль должен управлять распределенными данными.

Эти аспекты программирования находят отражение во всех модулях системы. Они могут проектироваться и разрабатываться в достаточной степени изолированно друг от друга, но, в конечном счете, должны быть объединены вместе на уровне исполняемого кода. Смешение различных аспектов во многом определяет сложность и ненадежность большинства существующих систем. Более того, в этой ситуации не поможет увеличение степени модульности системы без изменения самого принципа деления на модули. Для решения этой проблемы и была предложена новая парадигма аспектно-ориентированного программирования, где основной упор делается на выделение абстракций, моделирующих различные аспекты поведения системы, с их последующим автоматическим объединением для получения исполняемого кода.

Реализация различных аспектов функционирования системы может производиться с использованием разных шаблонов проектирования, методик, языков программирования и т.д. Для объединения различных аспектов в единое целое используется метаобъектный протокол. Протокол включает в себя понятие метаобъектов, которые, служат для управления поведением объектов и, таким образом, позволяют определять различные аспекты взаимодействия объектов: синхронизация, авторизация и т.д. Правильно спроектированный программный интерфейс метаобъектов делает возможным использование наследования и

полиморфизма для комбинирования, и построения новых метаклассов, реализующих различные аспекты поведения информационной системы.

Распределенные объектные ортогональные долговременные системы можно квалифицировать по способу расположения данных и выполнения программного кода на два класса: данные пересылаются к месту обработки и программный код передается к месту расположения данных. Возможен, естественно, и промежуточный вариант, где решение о том, что выгоднее, переслать данные или программу, принимается либо динамически на основании различных оптимизационных критериев, либо определяется статически путем явного указания.

Исследованы варианты выполнения запросов (программы). Показано, что выполнение программы или запроса в месте хранения долговременных объектов имеет ряд существенных достоинств: высокая производительность (возможен прямой доступ к данным), минимизация трафика по сети (от клиента передается только запрос и возвращается только результат его выполнения), невысокие требования к ресурсам рабочих станций клиентов. Однако эта модель малоэффективна в задачах, использующих навигационный подход, то есть осуществляющих обход коллекций долговременных объектов, а так же при работе со сложными сильносвязанными объектами.

На настоящий момент Интернет браузеры представляют собой целые программные системы, способные решать самые различные задачи. Все это послужило стимулом в пользу выбора модели выполнения кода на стороне клиента при построении распределенной объектной ортогональной долговременной Интернет системы.

Помимо традиционных вопросов, возникающих при построении любой долговременной системы, распределенная модель позволяет решать следующие задачи: выполнение кода методов клиентом позволяет лучше использовать ресурсы рабочих станций, но резко осложняет задачи синхронизации и поддержания корректности; необходимость пересылки объектов и сообщений приводит к значительному увеличению загрузки сети и поэтому производительность таких ортогональных долговременных распределенных систем ограничена пропускной способностью сети

Обозначенные принципы построения систем долговременного программирования легли в основу прототипа системы PWEB (Persistent WEB).

Показано, что при проектировании системы PWEB целесообразно остановится на модели активных клиентов, когда выполнение программного кода осуществляется на рабочих станциях под управлением стандартных программ просмотра всемирной сети. Результаты переносятся в хранилище по завершению транзакции. Такой подход, по мнению автора, наиболее подходит для разработки сложных программных систем, широко использующих интерактивный и навигационный доступ к информации, имеющей сложную и неоднородную структуру. Именно такими характерными чертами обладают современные Интернет приложения.

С технической точки зрения логическая структура системы реализуется следующим образом:

- на пользовательском уровне активные клиенты работают под управлением Интернет-браузеров таких, например, как Microsoft Internet Explorer или Mozilla Firefox, код активных клиентов реализуется на языке сценариев JavaScript, взаимодействие с сервером-хранилищем осуществляет по протоколу HTTP(s) с помощью асинхронных вызовов, что позволяет вести параллельную обработку информации;
- на уровне сервера приложений представлен Java 2 Enterprise Edition – совместимым сервером. Данный уровень отвечает за загрузку кода активных клиентов в Интернет-браузеры, а так же за обработку высокоуровневых запросов, с которыми обращаются активные клиенты;
- на уровне данных в системе PWEB представлен СУБД поддерживающей стандарт SQL-92 и построенной на сетях хранения данных в качестве хранилища информации.

Сеть Хранения Данных (СХД) – высокоскоростная коммутируемая сеть передачи данных, объединяющая серверы, рабочие станции, дисковые хранилища и ленточные библиотеки для решения следующих задач:

- консолидация (объединение) дисковых ресурсов, упрощается наращивание объемов хранения, снижается стоимость;
- обеспечение высокоскоростным подключением клиентских устройств на расстояния от нескольких метров до сотен километров;
- построение катастрофоустойчивых решений с территориально распределенными узлами обработки и хранения данных (централизованное

резервное копирование, репликация, процедуры восстановления и перехода на резервные узлы);

- обеспечение централизованного управления подсистемой хранения.

Выполнение какой-либо операции начинается непосредственно в браузере пользователя. В зависимости от того, какими объектами эта операция манипулирует, действие может быть выполнено полностью на стороне клиента, либо часть операции может быть автоматически делегирована на выполнение серверу-хранилищу. В последнем случае перед сервером ставится задача наиболее эффективного выполнения операции. Так как для выполнения метода объекта необходимо сначала загрузить объект с сервера, то эффективная работа системы невозможна без кэширования объектов на клиенте. При этом возникает проблема синхронизации содержимого кэш памяти различных клиентов. Задача синхронизации доступа к объектам является одной из наиболее сложных и критичных в распределенных долговременных системах с активным клиентом. Исследованы различные подходы к решению задачи синхронизации.

Для задания и расширения списка поддерживаемых стратегий синхронизации объектов в системе предложено использовать *метаобъекты*. Метаобъекты отвечают за поддержку параллельного доступа к долговременным объектам. Другой задачей метаобъектов является управление кэш памятью объектов загружаемых из хранилища, а так же управление доступом к объектам, т.е. если объект был загружен только для чтения, то метаобъекты сообщат об ошибке при любой попытке записи в объект, не дожидаясь сообщения об ошибке от хранилища. Исследованы возможные решения этих задач в системе.

Рассмотрены проблемы кэширования долговременных объектов в месте их использования, поскольку эта задача является критическим аспектом для эффективной работы ортогональных долговременных систем. Принято и обосновано решение о расположении в системе алгоритмов замещения в метаобъектах.

В третьей главе приводятся результаты разработки математических моделей для расчета характеристик обслуживания запросов при их параллельной обработке

В качестве модели для анализа обработки запросов исследована система с K ресурсами (в терминах СМО - приборами) и пуассоновскими потоками запросов, требующих для исполнения различного количества приборов. Здесь ресурсам

(приборам) соответствуют средства хранения данных, а потоки запросов к ресурсам образуются запросами к хранимым данным.

Интенсивность поступления запросов постоянна $\lambda_k, 0 < \lambda_k < \infty, k = \overline{1, K}$, т.е. все потоки стационарны и независимы в совокупности.

Длиной запроса считается количество необходимых приборов для его исполнения. Каждый запрос, поступивший в систему, характеризуется вектором $f = (f_1, f_2, \dots, f_K)$ фиксированной длины K , где $f_i = 1$ означает, что для исполнения запроса необходим i -й прибор, а $f_i = 0$ означает, что i -й прибор для исполнения запроса не требуется.

В качестве показателей производительности системы приняты U – коэффициент занятости приборов и w – среднее время ожидания в очереди.

Алгоритмы обслуживания предусматривают установление порядка обслуживания, который позволит максимально загружать приборы или минимизировать время ожидания в очереди. Это резко усложняет и сами алгоритмы, и задачу их исследования.

В связи с этим предложено упростить модель следующим образом: все K приборов объединяются в один общий ресурс, и считается, что k -запросу требуется q_k процентов ресурса, где q_k – целое, $0 < q_k \leq 100$. Длительность обслуживания k -запроса имеет экспоненциальное распределение с интенсивностью $\mu_k, 0 < \mu_k < \infty, k = \overline{1, K}$.

Исследована система с двумя входящими пуассоновскими потоками, интенсивности λ_1, λ_2 , буфером ограниченной емкости r , интенсивность обслуживания запросов 1-типа – μ_1 , запросов 2-типа – μ_2 . Допустимо параллельное исполнение запросов, при условии $(r_1 q_1 + r_2 q_2) \leq 100$, при этом возможно параллельное исполнение r_1 1-запросов и r_2 2-запросов.

Если на приборе достаточно свободного места, а в очереди ждут обслуживания запросы обоих типов, то на обслуживание поступает запрос, требующий больший процент ресурсов.

В качестве стохастической модели, описывающей функционирование СМО, рассмотрен марковский процесс $\{X(t), t \geq 0\}$, определенный на множестве состояний $X = \{X_{0000}, X_{1000}, X_{0100}, X_{1100}, \dots, X_{k_1 k_2 m_1 m_2}\}$.

Состояние процесса $X_{k_1 k_2 m_1 m_2}(t) = (k_1, k_2, m_1, m_2)$ для момента t значит, что в

данный момент времени t на обслуживании находится k_1 запросов 1-типа, k_2 запросов 2-типа, в очереди m_1 запросов 1-типа, m_2 запросов 2-типа. Число состояний процесса $\{X(t), t \geq 0\}$ конечно и все состояния процесса сообщаются. то $\{X(t), t \geq 0\}$ - эргодический процесс, и существуют финальные вероятности: $\lim_{t \rightarrow \infty} P\{X(t) = x\} = p_x > 0$, не зависящие от начального состояния процесса и совпадающие со стационарными вероятностями

Пусть $p_{k_1, k_2, m_1, m_2} = p(k_1, k_2, m_1, m_2)$ стационарная вероятность состояния (k_1, k_2, m_1, m_2) . С учетом дисциплины исполнения запросов, получена система уравнений для расчета вероятностей состояний.

Получены формулы, позволяющие вычислять значения основных показателей качества обработки запросов:

- средняя длина очереди: $Q = \sum_{K=0}^3 K \cdot s_K$, где $s_K = \sum_{k_1=0}^r \sum_{m_1=0}^r \sum_{m_2=0}^r p_{k_1, (K-k_1), m_1, m_2}$.

- среднее время ожидания в очереди запросов 1 и 2-типа: $w_1 = \frac{Q}{\lambda_1}$, $w_2 = \frac{Q}{\lambda_2}$.

- вероятность потерь π : $\pi = \tilde{\pi} + (p_{10r0} + p_{02r0}) \cdot \tilde{\lambda}_1$, где $\tilde{\lambda}_1 = \frac{\lambda_1}{\lambda_1 + \lambda_2}$.

Поскольку исследование системы при произвольном количестве входящих потоков ($K > 2$) представляет значительные трудности, для анализа систем большой размерности использован метод имитационного моделирования.

Построена имитационная модель, при помощи инструментальных средств GPSS WORLD. По имитационной модели произведены экспериментальные расчеты.

В четвертой главе приводятся варианты конкретной программной реализации предложенных методов построения системы на основе применения ортогонального программирования.

В приложении приводятся данные о практическом применении результатов диссертации при разработке программного обеспечения различного назначения.

ЗАКЛЮЧЕНИЕ. ОБЩИЕ ВЫВОДЫ

1. Обзор существующих теоретических и практических проблем, возникающих при создании ортогональных долговременных систем, показал необходимость использования более гибких подходов на этапе проектирования,

обеспечивающих сокращение этапов разработки и отладки программ.

2. На основе анализа существующих подходов, алгоритмов и современных теоретических достижений в управлении долговременными данными создана метамодель, которая позволяет реализовать любую из известных стратегий управления транзакциями и параллельным доступом к долговременным объектам, а также создавать специфичные для конкретной задачи модели поведения системы.

3. Разработаны математические модели для анализа процесса параллельной обработки запросов, позволяющие вычислять время обработки, вероятности потери запросов, определять параметры хранилища данных.

4. Разработан прототип системы PWEB обладающий базовыми возможностями объектно-ориентированной ортогональной системы долговременного программирования, позволяющими с использованием метаобъектного протокола расширять базовую функциональность системы до необходимого пользователю уровня.

5. На основе использования метаобъектного протокола, для решения проблемы объединения системы типов языка программирования, и модели данных СУБД предложен способ описания стабильных типов на языке XML. Как следствие ортогонального использования долговременных и кратковременных объектов, стандартный механизм связывания языка сценариев JavaScript был расширен механизмом связывания имени с сохраненным объектом в хранилище. Связывание также происходит непосредственно с прямым участием метаобъектов.

6. Эффективное использование метаобъектного протокола обуславливает необходимость тщательной разработки метаобъектного интерфейса и, таким образом, определять базовые классы. В соответствии с этим разработаны стандартные метаклассы для реализации оптимистической и пессимистической модели синхронизации. Разработан протокол взаимодействия хранилища с клиентом при фиксации транзакций.

7. Показано, что системы, подобные PWEB целесообразно строить по модульному принципу, с разбиением всех компонент системы на логически обособленные модули, взаимодействующие друг с другом посредством четко определенных интерфейсов. Это позволяет использовать различные реализации для каждого модуля с последующей настройкой под конкретные конфигурации для повышения производительности системы и удовлетворения специфических

требований целевых информационных систем. Эффективность различных алгоритмов и их пригодность для различных типов информационных систем можно оценить, экспериментируя с различными реализациями модулей, отвечающими за организацию кэш памяти, обработку транзакций, и т.д.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Зудин А.Б. Спецификация требований к долговременным объектным системам на основе сравнительного анализа моделей сохраняемости объектов // *Материалы международной конференции «Ломоносов-2003»*, М.: МГУ, 2003, с. 7.

2. Зудин А.Б. Системы долговременного программирования. // *Мир компьютерной автоматизации*. - М.: № 3, 2003. С. 50 – 56.

3. Зудин А.Б. Концепции интеграции языков программирования и баз данных // *Мир компьютерной автоматизации*. - М.: № 6, 2003. С. 34 – 39.

4. Зудин А.Б. Применение аспектно-ориентированного программирования в Web-системах со сложным клиентом // *Математическое и программное обеспечение информационных систем: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина – М.: Горячая линия - Телеком, 2007. - С. 86 – 92.*

5. Зудин А.Б. Методы хранения объектов в системах долговременного программирования // *Математическое и программное обеспечение вычислительных систем: Межвуз. сб. науч. тр. / Под ред. А.Н. Пылькина – М.: Горячая линия - Телеком, 2007. С. 92 – 95.*

6. Зудин А.Б. Системы долговременного программирования. // *Качество. Инновации. Образование. – М.: № 9, 2008. С. 17 – 20.*

7. Зудин А.Б. Модель обслуживания запросов в системах долговременного программирования // *Проблемы передачи и обработки информации в сетях и системах телекоммуникаций. Материалы 16-й международной научно-технической конференции. – Рязань, 2010. С. 83 – 85.*

8. Зудин А.Б. Модель доступа в системах долговременного программирования // *XLVI Всероссийская конференция по проблемам математики, информатики, физики и химии. Секции математики и информатики. Тезисы докладов. М.: РУДН, 2010. С. 53-54.*

9. Зудин А.Б. Система долговременного программирования // *Современные информационные компьютерные технологии (mсIT -2010). 2-я Международная научно-практическая конференция. Беларусь, Гродно, 2010. С. 57 – 61 .*

Подписано в печать 21.12.2010.

Формат 60×84/16. Бумага типографская № 2. Печать - ризография.
Усл. печ. л. 1,1 Тираж 100 экз. Заказ 1099.

Московский государственный институт электроники и математики
109028, Москва, Б.Трехсвятительский пер., 3.



Центр оперативной полиграфии
(495) 916-88-04, 916-89-25