РОССИЙСКАЯ АКАДЕМИЯ НАУК

ИНСТИТУТ ПРОБЛЕМ КИБЕРНЕТИКИ

На правах рукописи

ПОНОМАРЕНКО Вера Николаевна

ИССЛЕДОВАНИЕ ПРИНЦИПОВ УПРАВЛЕНИЯ ВНЕШНЕЙ ПАМЯТЬЮ И ТРАНЗАКЦИЯМИ В РЕЛЯЦИОННОЙ СУБД И РАЗРАБОТКА ПОДСИСТЕМЫ УПРАВЛЕНИЯ ПАМЯТЬЮ И СИНХРОНИЗАЦИЕЙ В РЕЛЯЦИОННОЙ СУБД

Специальность

05.13.11 - математическое и программное обеспечение вычислительных машин, комплексов, систем и сетей

А в т о р е ф е р а т

диссертации на соискание ученой степени кандидата физико-математических наук

Москва

- 1992 -

Работа выполнена в Институте проблем кибернетики АН России.

Научный руководитель -
кандидат физико-математических наук, старший научный сотрудник С. Д. Кузнецов.

Официальные оппоненты:
доктор физико-математических наук Л. А. Калиниченко,
кандидат физико-математических наук М. Р. Когаловский.

Ведущая организация - Институт прикладной математики им. М. В. Келдыша АН России.

Защита диссертации состоится " "29 октября 1992 г. в 14 " " часов на заседании специализированного совета К003.78.01 по присуждению ученой степени кандидата физико-математических наук в Институте проблем кибернетики АН России по адресу: 117312, Москва, ул. Вавилова, 37.

С диссертацией можно ознакомиться в библиотеке Института проблем кибернетики АН России.

Автореферат разослан "7 " сентября 1992 г.

Ученый секретарь
специализированного совета А. З. ИШМУХАМЕТОВ

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы.

В конце 70-х после появления ряда фундаментальных работ по теории реляционной модели данных и реализации в исследовательской лаборатории фирмы IBM экспериментальной реляционной СУБД System R начала нарастать популярность этого подхода к управлению данными. Реляционный подход к управлению базами данных обладает рядом преимуществ: простота и теоретическая обоснованность, ненавигационный доступ к данным и т. д. Основным недостатком реляционных систем долгое время считалась их недостаточная эффективность. По мере отработки алгоритмов и структур данных и развития техники оптимизации запросов стало возможным появление коммерческих реляционных СУБД, которые в настоящее время доминируют на мировом рынке.

Следует отметить особую роль языка реляционных баз данных SQL, который был разработан в рамках проекта System R, а в настоящее время стандартизован и поддерживается во всех современных СУБД.

Реляционная модель данных обладает рядом ограничений, среди которых основными являются обязательность атомарности значений атрибутов отношений, недостаточный уровень семантики, несоответствие непроцедурных языков запросов природе традиционных языков программирования. Ведется ряд работ, направленных на преодоление этих ограничений. В теоретическом плане это семантические и объектно-ориентированные модели данных. На практике имеется ряд постреляционных эспериментальных и даже коммерчески доступных СУБД.

Эти работы не снижают значимость реляционных систем и не уменьшают потребность в них. Во-первых, большинство СУБД, основанных на новых принципах, базируется на существующих реляционных системах (и, видимо, это будет продолжаться еще долгое время, пока не закончится период исследований). Во-вторых, многие авторитетные исследователи и разработчики новых СУБД провозгла-

шают требование их преемственности с реляционными системами баз данных.

Особенностью современного состояния в области реляционных СУБД является наличие большого числа теоретических и экспериментальных разработок и очень слабое использование этих разработок в реально используемых коммерчески доступных системах. Актуальной является разработка многопользовательской мобильной в среде открытых систем реляционной СУБД, в которой используются известные теоретические и экспериментальные результаты и новые идеи, направленные на повышение эффективности системы и увеличение надежности данных.

Цель диссертационной работы.

В основе любой современной многопользовательской СУБД лежит подсистема управления данными и транзакциями, обеспечивающая целостность базы данных, сериализацию транзакций и возможность восстановления состояния базы данных после различного вида сбоев.

Целью диссертационной работы являлось

- исследование принципов построения такой системы,
- анализ и выработка алгоритмов и структур данных,
- реализация системы в среде кластерной операционной системы (КЛОС) и в среде ОС UNIX.

Научная новизна.

Основными новыми результатами являются следующие:

- применение принципов КЛОС при выработке внутренней структуры ПУДТ;
- расширение интерфейса ПУДТ мощными "макро"-операциями;
- введение двух уровней синхронизации транзакций: логического, основанного на использовании предикатных синхронизационных захватов, и физического, основанного на захватах страниц;
- введение двух уровней журнализации: ведение логического журнала с записями уровня интерфейса ПУДТ и микрожурнала с записями о постраничных изменениях;

- введение понятия микрооперации для управления буфером оперативной памяти;

- применение индексной структуры в виде B-дерева для описания внешней памяти, занимаемой отношениями базы данных;

- обеспечение дополнительной временной индексной структуры (фильтра) и соответствующего набора операций для эффективного выполнения теоретико-множественных операций над отношениями и операций соединения отношений.

Апробация работы.

Основные результаты работы докладывались на

- на 5-й Всесоюзной конференции по базам данных и знаний (Львов, 1991);

- на научных семинарах отдела системного программирования (под рук. чл.-кор. АН России Иванникова В.П.) Института проблем кибернетики АН России (1988-1991).

Практическая ценность.

Работа начиналась в рамках проекта КЛЮС и была исходно ориентирована на то, чтобы обеспечить основу реляционной СУБД в среде этой операционной системы. В настоящее время ПУДТ перенесена в среду ОС UNIX. С использованием ПУДТ ведется работа по созданию свободно распространяемого SQL-сервера. Если учесть повсеместную потребность в SQL-сервере, работающем в UNIX-среде, а также высокую стоимость коммерчески доступных систем, можно оценить практическую важность диссертационной работы.

Публикации.

Основные результаты опубликованы в 3 работах, список которых помещен в конце реферата.

Структура диссертации.

Работа состоит из введения, пяти глав, заключения и списка литературы.

СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы диссертации, формулируются цели, отмечаются основные научные результаты и практическая значимость работы.

В первой главе приводится обзор научной литературы, относящейся к тематике диссертации.

Структурная организация современных СУБД является весьма сложной. Она должна поддерживать выполнение всех функций, которыми обладают развитые СУБД, а именно компиляцию запросов, управление транзакциями, обеспечение целостности БД, управление хранимыми данными, восстановление после сбоев и т. д. Структура каждой СУБД так или иначе отражает поставленные при ее разработке цели и выбранные решения. Проводится аналитический обзор структурных компонент таких известных реляционных СУБД, как System R и DB2.

Рассматривается современное состояние решений проблем управления внешней памятью и структуризации хранимых в БД данных. Обсуждаются подходы с применением различных видов индексов и кластеризации, а также проблемы управления буфером оперативной памяти.

Совместное выполнение нескольких корректных самих по себе транзакций может продуцировать ошибочный результат при отсутствии надлежащего механизма управления транзакциями. Ключевой проблемой управления транзакциями в СУБД является обеспечение сериального выполнения смеси транзакций. Обсуждаются пессимистические и оптимистические методы сериализации, рассматриваются преимущества и недостатки методов синхронизационных захватов и временных меток.

Обсуждаются проблемы физической и логической целостности БД. Логическая целостность БД обычно поддерживается с помощью запоминания в каталогах базы данных ранее сформулированных ог-

раничений целостности. Физическая целостность БД обычно обеспечивается соблюдением протокола упреждающей записи в журнал при использовании в СУБД журнализации.

Надежность хранения данных в БД является основным требованием к СУБД. Это означает, что СУБД обязана поддерживать средства восстановления БД после всевозможных сбоев. Обеспечение такой надежности достигается с помощью регистрации всех производимых над данными изменений в журнале.

Современная СУБД поддерживает средства восстановления состояния БД после любых сбоев, которые могут быть вызваны сбоями отдельных транзакций (например, деление на ноль в прикладной программе, вызвавшей выполнение данной транзакции), сбоями процессора (мягкие сбои) или сбоями внешних носителей, на которых расположена БД (жесткие сбои). Обсуждаются протоколы восстановления после сбоев с применением журнализации и теневого механизма.

Вторая глава содержит описание общей организации реляционной СУБД, основой которой служит ПУДТ. Особенностью такой СУБД, учитываемой при организации ПУДТ, является ее SQL-ориентированность. Рассматриваются принципы именования объектов базы данных. Обосновываются и описываются внутренняя структуризация ПУДТ и ее внешний интерфейс.

ПУДТ, которой посвящена эта работа, рассчитана на использование в СУБД, ориентированных на использование языка SQL. Этот язык соответствует реляционной модели данных и оперирует, в основном, в терминах отношений и составляющих их кортежей.

Выбирая способ реализации SQL, который представляет на уровне пользователя интерфейс с СУБД, важно выбрать правильный подход к структуризации системы. Очевидно, что для реализации SQL должен существовать набор внутренних средств работы с БД. Эти средства должны обеспечивать управление данными на внешней памяти, надежность хранения БД и необходимую синхронизацию, если реализация должна поддерживать режим мультидоступа. Важно правильно решить, какие действия должны производиться в рабочей программе, полученной после компиляции предложения SQL, а какие

могут выполняться стандартным образом внутри подсистемы поддержки.

Такой системой поддержки и является подсистема управления данными и транзакциями. Для ее реализации был выбран подход с журнализацией изменений БД и поддержкой двухфазного протокола синхронизационных захватов.

В обычном режиме работы ПУДТ состоит из шести постоянно присутствующих в подсистеме кластеров: Администратора, Синхронизатора, Логического Журнала, Микрожурнала, Буфера и Сортировщика. Количество кластеров-Транзакций в подсистеме соответствует числу одновременно выполняющихся транзакций в подсистеме.

Для надлежащего функционирования СУБД должна располагать некоторой системной или справочной информацией. Обычно совокупность такой информации носит название каталога, который является системной базой данных и содержит информацию о разнообразных объектах, представляющих интерес для самой системы. Пользователь БД оперирует в своих запросах именами объектов, таких как таблицы, столбцы, индексы, однако именование объектов является лишней информацией для ПУДТ. Соответствие между именами обектов и их физическим расположением в БД возлагается на верхний уровень системы, который находит эту информацию в системном каталоге.

ПУДТ в своей работе пользуется лишь одним системным отношением - отношением-каталогом, в котором предусматривается строка для каждого отношения данных. Такая интерпретация отношения-каталога дает возможность ПУДТ обращаться с ним, как с обычным отношением.

В интерфейс описываемой подсистемы входят операции сканирования отношения. Имеется три способа сканирования отношения. Отношение можно сканировать последовательно в порядке, предопределенном ПУДТ. Можно сканировать отношение в порядке, задаваемом любым существующим индексом на этом отношении. Наконец, можно сканировать отношение в порядке, задаваемом существующим на этом отношении фильтром.

С открытым сканом можно выполнять следующие операции:

- найти следующий кортеж,
- выбрать из текущего,
- запомнить текущую позицию сканирования,
- сделать запомненную позицию текущей,
- удалить кортеж, задаваемый текущей позицией сканирования,
- модифицировать кортеж, задаваемый текущей позицией сканирования,
- закрыть сканирование.

В набор "макро"-операций входят следующие операции:

- отфильтровать отношение,
- удалить кортежи, удовлетворяющие условию,
- модифицировать кортежи, удовлетворяющие условию,
- вставить в отношение набор кортежей другого отношения, удовлетворяющих условию,
- породить отсортированный фильтр в соответствии с заданным условием.

Операция вставки кортежей в отношение:

- вставить кортеж.

Сортировка и теоретико-множественные операции над отсортированными временными объектами:

- отсортировать временный объект.

Допускаются теоретико-множественные операции над отсортированными временными объектами: объединение, пересечение и разность. Для отсортированных временных отношений допускается операция эквисоединения, т.е. соединение с предикатом равенства.

Операции, связанные с изменением схемы БД:

- создать постоянное или временное отношение,
- создать фильтр,
- создать индекс,
- уничтожить временный объект,
- уничтожить постоянное отношение,
- уничтожить индекс.

Операция изменения схемы отношения:

- добавить поля к существующему отношению.

Операции управления прохождением транзакций:

- установить контрольную точку,
- откатить транзакцию до указанной контрольной точки.

Операция завершения транзакции:

- завершить транзакцию.

Интерфейс подсистемы реализуется в кластере-транзакции.

В третьей главе рассмотрены структуры внешней памяти, используемые в ПУДТ для организации баз данных и обеспечения эффективного доступа к данным. В этой же главе описываются алгоритмы управления буферами оперативной памяти - основы эффективной работы любой СУБД и соответствующий механизм синхронизации транзакций нижнего уровня.

Для работы с внешней памятью используются средства, предоставляемые файловой системой, что позволяет пользоваться такими средствами файловой подсистемы, как хранение справочников и файлов в виде иерархической структуры и обращение к файлам через их имена в архиве.

БД, с которой манипулирует ПУДТ, располагается в одном или нескольких сегментах - файлах внешней памяти со страничной организацией.

Все файлы-сегменты одной БД должны быть размещены в одном справочнике (его имя является именем БД и иметь в нем стандартные предопределенные имена (номера сегментов).

Различаются три типа сегментов: сегменты журналов, рабочий сегмент и восстанавливаемые сегменты. Для каждого типа сегментов ПУДТ поддерживает разные структуры хранения.

Основная часть БД - постоянные отношения и индексы - хранится в восстанавливаемых сегментах. При этом каждое отношение располагается в одном сегменте, в этом же сегменте находится его описатель и все индексы, определенные на данном отношении.

Рабочий сегмент БД используется для размещения временных объектов транзакций (временных отношений и фильтров). При сортировке используются буфера из пула ПУДТ и рабочая внешняя память, в качестве которой используется память того же рабочего сегмента.

Сегменты журнала служат для хранения информации об изменениях в БД. Поддерживается два журнала - логический журнал, в который заносятся записи о выполнении операций уровня интерфейса ПУДТ (обычных, а не "макро"-операций), и микрожурнал, содержащих записи о модификации страниц. Логический журнал существует до момента копирования состояния БД (он копируется вместе с остальными сегментами), микрожурнал начинает заполняться заново после фиксации на внешней памяти содержимого буферов ПУДТ.

Для каждой страницы отношения поддерживается описатель, содержащий номер страницы в сегменте и размер свободной памяти в этой странице. Описатель существует только для тех страниц сегмента, в которых размещены кортежи отношений. Доступ к описателю производится с помощью иерархической структуры, организованной в виде B-дерева. Описатели страниц каждого отношения образуют поддеревья общего B-дерева; полный ключ описателя состоит из номера отношения и номера страницы в сегменте.

Организация B-дерева описателей страниц очень близка к организации индексов. Для работы с этими структурами используется общий набор программ.

Каждое отношение идентифицируется tid'ом своего кортежа - описателя в отношении-каталоге. Схема отношения каталога предопределена и фиксирована, и поэтому это служебное отношение не нуждается в наличии описателя. Отношение-каталог тем самым не имеет идентификатора и потому к нему невозможен доступ через интерфейс ПУДТ.

B-деревья индексов включают страницы двух типов: промежуточные и листовые. В промежуточных страницах находятся значения ключей и ссылки на страницы непосредственно подчиненного уровня. Листовые значения содержат значения ключей и для каждого значения - список tid'ов кортежей, упорядоченный по возрастанию значений tid'ов. Листовые страницы связаны в однонаправленный список, соответствующий возрастанию ключа.

Ключи индексов могут быть простыми (соответствовать одному полю отношения) и составными (соответствовать нескольким полям отношения в их заданной последовательности). Для всех допусти-

мых типов простых ключей определено отношение порядка и обеспечивается набор функций сравнения значений. Порядок составных ключей определяется лексикографически на основе порядков составляющих их простых значений. Допускается попадание в индекс ключей с неопределенными значениями.

Страница данных предназначена для хранения кортежей отношения. Кортеж располагается только в одной странице. Уникальный идентификатор кортежа - tid - состоит из номера страницы в сегменте и идентификатора кортежа внутри страницы (tid идентифицирует кортеж в пределах данного сегмента). Внутренний идентификатор кортежа (в пределах страницы) есть номер указателя на кортеж; этот указатель хранится в той же странице. Tid кортежа не может меняться, пока этот кортеж существует.

Что касается структур данных, располагаемых в рабочем сегменте, то с ними дела обстоят проще, чем в восстанавливаемых сегментах. Служебная информация, подобная B-дереву описателей страниц в восстанавливаемом сегменте, в рабочем сегменте не поддерживается. Все необходимые описатели рабочих областей транзакций находятся в оперативной памяти соответствующих кластеров.

Структура страниц, содержащих кортежи временных отношений, полностью подобна структуре соответствующих страниц восстанавливаемых сегментов.

Структура страниц, содержащих фильтры, очень проста: это просто линейный список tid'ов.

ПУДТ поддерживает пул буферов - сегментов в смысле КЛОС (в среде ОС UNIX также используются разделяемые сегменты).

Главное назначение пула буферов - уменьшение числа обменов с внешней памятью за счет удержания в оперативной памяти копий страниц сегментов. В этом смысле пул буферов - это программно организованный кэш для доступа к БД.

Управлением пулом буферов ПУДТ ведает специальный компонент ПУДТ - кластер-Буфер.

С точки зрения синхронизации кластер-Буфер поддерживает некоторую разновидность двухфазного протокола захватов на уров-

не страниц данных.

Кластерная операционная система дает возможность реализовать кластер-Буфер таким образом, что он удовлетворяет запросы на страницу от всех кластеров-Транзакций, ранее ее захвативших (конечно, при вопросе о разделяемой памяти имеются в виду только захваты на чтение). Аналогичные возможности обеспечиваются механизмом управления разделяемой памятью ОС UNIX.

Причины, вызвавшие потребность разделения синхронизационных захватов страниц и запросов буферов, содержащих страницы, связаны с подходом к обеспечению физической целостности БД после мягких сбоев, сопровождающихся потерей оперативной памяти. При этом содержимое части буферов может оказаться вытолкнутым на внешнюю память, а содержимое другой части просто пропадает. В результате после мягкого сбоя содержимое БД может придти в рассогласованное состояние.

Для того, чтобы можно было восстановить физическую согласованность БД, используется техника сохранения информации о постраничных изменениях в микрожурнале. Перед обновлением микрожурнала (т.е. моментом, когда он начинает заполняться заново) необходимо вытолкнуть на внешнюю память содержимое всех буферов, в которых происходила запись.

Поскольку операция является довольно крупной единицей, и не хотелось бы производить откат операции целиком при мягких сбоях (под операцией понимается либо разовая операция из интерфейса ПУДТ - "найти следующий", "вставить кортеж" и т.д., либо эквивалентные им части массовых операций.), вводится понятие микрооперации.

Микрооперацией называется часть операции, заключенная между двумя моментами времени, в каждый из которых операция может (но не обязательно должна) отказаться от всех своих захватов. Эти моменты являются своего рода контрольными точками операции. Откат операции всегда производится на начало текущей микрооперации. Кластер-Транзакция обязан обеспечивать возможность возобновления работы с момента начала текущей микрооперации.

Чтобы не накладывать какие-либо ограничения на порядок

разбиения операций на микрооперации, запросы и захваты страниц в кластере-Буфере разделяются.

Общий подход к политике замещения буферов в пуле заключается в учете старения буферов, т.е. первым кандидатом на замещение является буфер, который наиболее давно не запрашивался кластерами-Транзакциями. При этом учитываются предпочтения некоторых буферов на предмет сохранения их содержимого.

Что касается распознавания синхронизационных тупиков в кластере-Буфере, то при реализации было выбрано решение, основанное на контроле времени ожидания удовлетворения захвата, поскольку число захватов страниц должно быть невелико, и вероятность возникновения тупика незначительна.

Четвертая глава посвящена синхронизации транзакций верхнего уровня, логической синхронизации транзакций. Особенностью основанного на двухфазном протоколе синхронизационных захватов механизма синхронизации является то, что захватываются не физические объекты базы данных, а логические условия - предикаты. Сама идея предикатных захватов не является новой, но несмотря на очевидные преимущества, предикатные захваты мало где используются по причине сложности реализации. В ПУДТ удалось добиться эффективной реализации этого механизма с поддержанием аппарата распознавания и разрушения тупиков.

В данной системе синхронизация транзакций происходит на уровне ПУДТ, в которой ничего не известно про предикаты исходных запросов. На этом уровне в роли "языка запросов" выступает набор операций интерфейса ПУДТ, а в этом интерфейсе допускаются только очень простые предикаты, представленные в виде логических формул, в которых через конъюнкцию соединены простые условия на поля отношения.

Логической синхронизацией управляет в подсистеме кластер-Синхронизатор. Захваты объектов в ПУДТ могут выполняться в двух режимах: совместном (на чтение) и монопольном (на обновление).

Итак, область логического захвата задается предикатом, т.е. для указанного отношения для некоторых атрибутов указывается диапазон значений. Вводятся понятия общего и элементарного

захвата. Элементарный захват является конъюнкцией диапазонов значений атрибутов отношения. Общий захват является дизъюнкцией элементарных захватов.

Вводятся понятия "узкого" и "широкого" захватов, объясняющиеся следующими причинами. В начале сканирования область сканирования захватывается на чтение. Чтобы избежать захвата на изменение всей области сканирования при первой же попытке транзакции удалить или изменить текущий кортеж, транзакция порождает узкий захват, т.е. захват предиката, в точности характеризующего текущий кортеж. Первые несколько изменений делаются узкими захватами. В большинстве случаев, видимо, этих узких захватов будет не очень много. При достижении некоторого критического числа узких захватов производится захват на обновление всей области. Решение вопроса о том, когда именно нужно переходить от узких захватов к широкому, возлагается на кластер-Транзакцию. Такая последовательность захватов является компромиссом между сильным размножением узких захватов, что может привести к перегрузке таблиц синхронизатора, и монополизации отношения, из-за чего может уменьшиться степень параллельности выполнения транзакций.

Кластер-Синхронизатор проверяет каждый элементарный захват на совместимость с удовлетворенными на данный момент захватами. В зависимости от того, могут сейчас быть выполнены эти элементарные захваты или нет, они включаются соответственно в списки удовлетворенных или ждущих захватов.

Приводятся подробные алгоритмы обработки поступления и снятия захватов.

Описываются примитивы кластера-Синхронизатора.

Для распознавания и разрушения тупиков Кластер-Синхронизатор поддерживает граф ожидания, вершинами которого являются транзакции. Поскольку просмотр общего захвата останавливается на первом блокированном элементарном захвате, в каждый момент времени любая транзакция может ждать только одну транзакцию, т.е. исходящее ребро из вершины-транзакции в графе ожидания только одно.

Описываемая схема поддержки графа ожидания позволяет производить проверку на тупик сразу при появлении ждущего захвата. На транзакции, у которой только что появился ждущий захват, начинается движение по исходящим ребрам графа ожидания до замыкания кольца или до вершины, у которой отсутствует исходящее ребро. Замыкание кольца означает, что в системе образовался тупик, который немедленно разрушается. Для этого делается второй обход тупикового кольца теперь уже с выбором жертвы для отката. Ею становится транзакция со входящим ребром, имеющим минимальную цену отката. Цена отката транзакции находится в зависимости от количества произведенных ею обменов с внешней памятью и вычисляется как разность между текущей стоимостью транзакции и стоимостью в найденной контрольной точке.

В пятой главе рассматривается механизм журнализации. Вместе с механизмом синхронизации журнализация обеспечивает поддержание целостного состояния базы данных. Особенностью основанного на протоколе упреждающей записи в журнал механизма журнализации является разделение журнала на две части: долговременного логического журнала, содержащего записи уровня интерфейса ПУДТ, и более часто обновляемого микрожурнала с записями о постраничных изменениях. Обосновываются и описываются протоколы журнализации, завершения транзакций и восстановления на основе содержимого журналов базы данных после разного рода сбоев.

Приводятся примитивы кластеров логического журнала и микрожурнала. Описываются структуры страниц этих журналов.

В заключении приводятся основные результаты работы, описывается текущее состояние проекта и обсуждаются возможные перспективы.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ

В диссертационной работе получены следующие основные результаты:

1. На основе применения принципов и методологии кластерной операционной системы (КЛОС), следуя идеям объектно-ориентированного подхода в программировании, выработана структура подсистемы управления данными и транзакциями (ПУДТ), предназначенной для реализации SQL-ориентированной СУБД. Подсистема состоит из статически определенного набора функционально-ориентированных активных кластеров, взаимодействующих путем обмена сообщениями.

2. Базирование на простых механизмах обмена сообщениями и использовании разделяемых сегментов оперативной памяти, ограниченные требования к внешней среде и использование для программирования стандартного варианта языка Си позволили достичь высокого уровня мобильности ПУДТ. Задуманная как машинно-независимая система в среде КЛОС, ПУДТ легко и естественно была перенесена в среду ОС UNIX, причем в UNIX-реализации используются только те средства ядра ОС UNIX, которые присутствуют во всех современных вариантах этой ОС.

3. В целях эффективной реализации на основе ПУДТ SQL-ориентированной СУБД в интерфейс ПУДТ наряду с традиционными для таких систем покортежными операциями введены так называемые "макро"-операции, позволяющие в ряде случаев за одно обращение к ПУДТ выполнить целиком операцию реляционной алгебры. Это не только позволяет сократить число обменов сообщениями во время выполнения оператора SQL, но и дает возможность добиться максимально эффективного его выполнения за счет полного использования механизма буферизации ПУДТ.

4. Для обеспечения сериализации транзакций на должном уровне их изолированности в ПУДТ применен двухфазный протокол синхронизационных предикатных захватов с обнаружением и разрушением возможных синхронизационных тупиков. Для увеличения степени асинхронности транзакций при синхронизации доступа к стра-

ницам базы данных используется дополнительный внутренний механизм синхронизации с кратковременными захватами страниц. Поддержание этого механизма основано на введенном понятии микрооперации - части операции уровня интерфейса ПУДТ, для которой требуется сохранение монопольных захватов страниц.

5. Для сокращения объема журнала изменений базы данных и упрощения его структуры введены два уровня журнализации: в долговременный логический журнал заносятся записи уровня интерфейса ПУДТ; в более часто обновляемый (и не архивизируемый) микрожурнал включаются записи о постраничных изменениях базы данных. Тщательное следование протоколу упреждающей записи в журналы, согласование последовательности синхронизации операций и журнализации изменений и надежное поддержание журнальных файлов обеспечивают возможности индивидуальных откатов транзакций и восстановления состояния базы данных после различного рода аппаратных и программных сбоев.

6. Эффективный просмотр отношений базы данных и распределение памяти в файлах, хранящих отношения, с поддержанием в случае необходимости динамической кластеризованности отношений обеспечиваются за счет применения специальной индексной структуры для описания внешней памяти, занимаемой отношениями базы данных. Аналогичный механизм с тем же самым набором программ используется для организации дополнительных путей доступа (индексов) к отношениям.

7. Применение индексной структуры в виде B-дерева для описания внешней памяти, занимаемой отношениями базы данных.

8. Обеспечение дополнительной временной индексной структуры (фильтра) и соответствующего набора операций для эффективного выполнения теоретико-множественных операций над отношениями и операции соединения отношений.

9. Подсистема управления данными и транзакциями была реализована в опытном варианте в первой версии КЛОС на ПЭВМ "Электроника 85". Малые ресурсы этой ПЭВМ позволили использовать ее только для отладки и проверки правильности основных решений. В дальнейшем ПУДТ вместе с КЛОС была перенесена на более

мощную рабочую станцию "Беста", где была завершена комплексная отладка и были проведены работы по оценке эффективности подсистемы.

10. Перенос ПУДТ в среду ОС UNIX System V был также произведен на рабочей станции "Беста". В настоящее время произведен еще один перенос ПУДТ на ЭВМ типа VAX, работающую под управлением ОС ULTRIX (разработанный на фирме DEC аналог UNIX BSD 4.3). Это продемонстрировало мобильность ПУДТ в разнородной UNIX-среде.

11. В настоящее время с использованием ПУДТ реализуется проект свободно распространяемого SQL-сервера, предназначенного для использования в локальных сетях UNIX-машин. При работе в среде ОС UNIX для доступа к ПУДТ используется механизм гнезд (sockets). Это позволяет использовать ПУДТ как в локальном варианте (когда прикладные программы и ПУДТ работают в одном компьютере), так и в удаленном режиме (когда прикладные программы обращаются к компьютеру-серверу через средства локальной сети). Для прикладных систем, естественно, обеспечивается прозрачность доступа.

* * *

Особую признательность автор выражает научному руководителю С.Д. Кузнецову за постоянную помощь в работе и написании диссертации и В.П. Иванникову за внимание и поддержку проекта.

В разработке проекта ПУДТ, выработке основных алгоритмов, протоколов и структур данных участвовали И.Б. Бурдонов, С.Д. Кузнецов, П.В. Логвин, С.В. Шпекторов, В.Н. Юдин. В практическом написании программ принимали участие Н.В. Игнатьева и С.В. Шпекторов. Всем им автор приносит свою искреннюю благодарность.

Автор также искренне благодарит А.С. Косачева, Г.В. Копытова и Е.Г. Березина за консультации по операционной системе КЛЮС и приятную совместную работу.

ПУБЛИКАЦИИ ПО МАТЕРИАЛАМ РАБОТЫ

1. И.Б. Бурдонов, Н.В. Игнатьева, С.Д. Кузнецов, В.Н. Пономаренко, С.В. Шекторов. Функции и организация подсистемы управления памятью и синхронизацией реляционной СУБД. В сб. "Вопросы кибернетики. Программное обеспечение высокопроизводительных систем". М: изд-во НСК АН СССР. - 1991. - с. 71-97.

2. С.Д. Кузнецов, В.Н. Пономаренко. Модульная мобильная система управления данными и транзакциями. Труды 5-ой Всесоюзной конференции по базам данных и знаний. - Львов, 23-27 сент. 1991, УМиМ, 1991, N 7, с. 37-45.

3. Н.В. Игнатьева, С.Д. Кузнецов, В.Н. Пономаренко. Реализация подсистемы управления данными и транзакциями в ОС КЛОС. В сб. "Вопросы кибернетики. Программное обеспечение высокопроизводительных систем". М: изд-во НСК АН СССР. В печати.