На правах рукописи

ПЫСТОГОВ Сергей Васильевич

СУБД ПОЛНООБЪЕКТНЫХ КАРТОГРАФИЧЕСКИХ СЦЕН С АССОЦИАТИВНОЙ ЗАЩИТОЙ НА КЛАСТЕРНОЙ ПЛАТФОРМЕ

Специальность:
05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени кандидата технических наук

Казань – 2019

Работа выполнена на кафедре компьютерных систем ФГБОУ ВО «Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ» (КНИТУ-КАИ)

Научный руководитель:	Доктор физико-математических наук, профессор **РАЙХЛИН Вадим Абрамович** профессор кафедры компьютерных систем КНИТУ-КАИ
Официальные оппоненты:	Доктор физико-математических наук, профессор **СОКОЛИНСКИЙ Леонид Борисович** проректор по информатизации, заведующий кафедрой системного программирования ФГАОУ ВО «Южно-Уральский государственный университет (НИУ)»
	Кандидат технических наук **ГУСЕНКОВ Александр Михайлович** доцент кафедры технологий программирования Института вычислительной математики и информационных технологий ФГАОУ ВО «Казанский (Приволжский) федеральный университет»
Ведущая организация:	ФГБОУ ВО «Смоленский государственный университет», г. Смоленск

Защита диссертации состоится 14 июня 2019 г. в 9^{00} часов на заседании диссертационного совета Д 212.288.07 на базе ФГБОУ ВО «Уфимский государственный авиационный технический университет» по адресу: 450008, г. Уфа, ул. К. Маркса, 12.

С диссертацией можно ознакомиться в библиотеке ФГБОУ ВО «Уфимский государственный авиационный технический университет» и на сайте *www.ugatu.su*.

Автореферат разослан «___» _____ 2019 года.

Ученый секретарь
диссертационного совета
докт. техн. наук, доцент И.Л. Виноградова

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. В настоящее время во всем мире широко используются средства пространственного анализа данных (анализа сцен) различными структурами и учреждениями. Под ***анализом сцен***[1] подразумевается одна из задач распознавания образов, когда интересуются лишь укрупненным описанием того, что представлено на изображении, в терминах «объекты – координаты». Использование пространственных данных уже давно вышло за пределы военных и промышленных ведомств. Объемность таких данных, достигающая десятков гигабайт для сцен среднего размера, делает необходимым использование при их анализе высокопроизводительных параллельных систем.

Полнообъектная картографическая сцена **(ПКС) –** участок карты (изображение) с возможным присутствием на нем всех типов объектов – точечных, линейных, площадных. Векторная модель данных отображает объекты карты в виде точек, линий и плоских замкнутых фигур. Сцена может быть разделена на *тематические слои* в соответствии с той или иной предметной областью. Тематические слои могут содержать *конфиденциальные сведения.*

Особенностям построения пространственных баз данных уделяется в настоящее время серьезное внимание. Для организации хранения и управления пространственными данными в СУБД широко применяется серверное программное обеспечение *ArcSDE* в составе *ArcGIS Server*. Ведущие места среди СУБД, обладающих встроенными механизмами защиты баз данных, занимают СУБД *Oracle, Microsoft SQL Server, Sybase Server*.

Разработанный в диссертации исследовательский прототип ассоциативно-защищенной СУБД предназначен для работы с пространственными данными, хранящимися в замаскированном виде. Система реализована на платформе вычислительных кластеров, использует в качестве базовой СУБД MySQL. Для маскирования данных используются оригинальный двумерно-ассоциативный механизм, который является аналогом исторического трафаретного способа стеганографии (роль трафарета играет инверсная матрица маски размерами mxn, единицы которой отмечают позиции сохраняемых бит в двоичном эталоне). Разработка системы велась с учетом требований обеспечения технологической независимости страны в области СУБД [2]:

1. Согласно постановлению Правительства № 1236 от 16 ноября 2015 г. установлен запрет на допуск программного обеспечения, происходящего из иностранных государств, для обеспечения государственных и муниципальных нужд.

2. Чтобы не оказаться в догоняющей позиции, новые отечественные СУБД следует создавать на базе готовых СУБД с открытым кодом и свободной лицензией, поддерживаемых международным сообществом.

3. Вновь создаваемые в России СУБД должны быть защищены.

К этому целесообразно добавить следующее.

4. В работе[3] выявлены существенные преимущества использования набора масок в качестве ключа распознавания имен и координат (их десятичных кодов, представленных в виде набора бинарных матриц) объектов картографических сцен по сравнению с использованием для целей защиты КС рекомендованного к применению в нашей стране шифра ГОСТ 28147-89, при сохранении доказуемой стойкости защиты:

---

[1] Дуда Р., Харт П. Распознавание образов и анализ сцен. – М.: Мир, 1976.

[2] Российская отрасль СУБД продвигается на «слонах» // Connect. 2017. №5-6. С.34

[3] Гибадуллин Р.Ф. Система баз данных картографии с ассоциативной защитой /Автореф. дис. ... канд. техн. наук. Уфа, 2011. – 16 с.

- удвоение скорости обратных преобразований скрываемых данных;
- допускается искажение до 3% хранимых и передаваемых бит вместо 1,5% – для ГОСТ 28147-89. Возможен дальнейший рост помехоустойчивости за счет введения избыточности.

5. Ранее созданная система[3] допускала работу только с точечными объектами. Необходимость сокрытия и дальнейшего распознавания протяженных объектов (линейных и площадных) определена задачами обороны, экономики и др.

В силу предыдущего, развитие методов и алгоритмов программной организации отечественных СУБД ПКС, реализующих разработанные в упомянутой работе[2] элементы теории ассоциативной стеганографии и используемых, в частности, в качестве инструментального средства при проведении исследований по развитию этой теории, является актуальной задачей.

Степень разработанности темы. Вопросами представления и обработки пространственных данных занимаются ученые В.Я. Цветков, А.М. Берлянт, B. Plewe, M.DeMers. Обоснованность хранения таких данных в защищенном виде приводится в работах Л.К. Бабенко, О.Б. Макаревича, Y.Dakruri. Особенностям организации пространственных баз данных уделено серьезное внимание в работах Б.Б. Серапинаса, И.К.Лурье, J.Ullman, S.Shekhar, S.Chawla, N.Adam, J.Albrecht. Вопросы построения высокопроизводительных систем изучены в трудах Вл.В.Воеводина, В.В.Корнеева, В.П.Гергеля, R.Hockney, а вклад в развитие параллельных систем баз данных внесли ученые Л.Б.Соколинский, С.Д.Кузнецов, М.Р.Когаловский, M.Stonebraker, D.DeWitt.

Объект исследования. СУБД полнообъектных картографических сцен с ассоциативной защитой.

Предмет исследования. Методы, алгоритмы и программные реализации системы управления базами данных полнообъектных картографических сцен с ассоциативной защитой.

Цель диссертационной работы. Повышение функциональности (достижение полнообъектности при некритичности к размерам объектов) ассоциативно-защищенных СУБД КС с применением кластерных платформ.

Решаемые задачи

1. Разработка метода организации (схемы) ассоциативно-защищенной БД ПКС, универсальной к типам картографических объектов;
2. Разработка метода, алгоритмов и программ формирования таких БД на вычислительном кластере, в частности, – для целей тестирования;
3. Разработка метода обработки запросов и на его основе алгоритма и программ организации серверной части ассоциативно-защищенной СУБД ПКС;
4. Исследование динамики процессов при обработке пакетов запросов в разработанных программных архитектурах серверной части системы и формирование практических рекомендаций по выбору наиболее подходящей архитектуры и улучшению программной организации;
5. Разработка архитектуры, алгоритма и программ клиентской части системы для взаимодействия с сервером управления БД ПКС.

Основные научные результаты, выносимые на защиту

1. Метод организации (схема) базы данных ассоциативно-защищенной СУБД ПКС, основанный на разделении тематических слоев для разных типов объектов и их единообразном представлении, и на его основе алгоритм формирования БД ПКС.

2. Метод и на его основе алгоритм стохастической генерации тестовых баз данных, основанные на выделении в сцене прямоугольной области для каждого запроса представительского теста.

3. Метод и на его основе алгоритмы и программы выполнения запросов в серверной части СУБД ПКС для двух возможных режимов её работы, основанные на параллельной обработке таблиц ассоциативно-защищенной БД ПКС без их предварительного раскрытия, в том числе с выборкой частей линейных и площадных объектов для селективных запросов.

4. Результаты исследования динамики работы серверной части системы для двух режимов работы.

5. Алгоритм и программный прототип клиентской части системы, основанные на предварительной обработке запросов пользователя и дополнительной обработке полученных от сервера результатов.

Научная новизна работы

1. Предложен метод организации (схема) БД ПКС с ассоциативной защитой, основанный на разделении тематических слоев для разных типов объектов с выделением своего слоя для любого линейного и площадного объекта, что, в отличие от известного[3], позволяет снять ограничения на размеры протяженных объектов и единообразно хранить объекты разного типа.

2. Предложен метод генерации тестовых баз данных защищенных ПКС, основанный на выделении в сцене прямоугольной области для каждого запроса представительского теста, что, в отличие от тестов TPC (Transaction Processing Performance Council), позволяет проводить тестирования производительности СУБД ПКС.

3. Предложен метод обработки селективных запросов к БД ПКС с ассоциативной защитой по выборке частей линейных и площадных объектов, основанный на использовании единого формата данных для всех типов объектов, что, в отличие от известных методов, позволяет проводить выборку точечных объектов и частей протяженных объектов с применением платформ вычислительных кластеров без предварительного «раскрытия» всей БД.

Научная значимость

1.Разработка натурной модели (прототипа, воспроизводящего реальные системные ситуации) серверной части ассоциативно-защищенной СУБД ПКС для целей исследований.

2.Установление по результатам экспериментов на этой модели предпочтительности выбора так называемого монокластерного режима работы указанной СУБД.

Практическая ценность работы заключается в разработке практических рекомендаций по созданию ассоциативно-защищенной СУБД ПКС на платформе вычислительных кластеров. Созданный исследовательский прототип системы может быть использован как действующая платформа для параллельной обработки конфиденциальных картографических данных, хранимых в маскированном виде и передаваемых по открытым каналам связи, дальнейших исследований вопросов анализа таких данных, и изучения вопросов организации специализированных систем управления базами данных в учебном процессе вузов.

Обоснованность и достоверность результатов диссертации

Достоверность подтверждена результатами экспериментальных исследований, проведенных на натурной модели, которая разрабатывалась с применением рекомендованных программных инструментальных средств.

Соответствие работы специальности ВАК 05.13.11

Работа отвечает следующим пунктам паспорта специальности 05.13.11 (в квадратных скобках указаны соответствующие позиции диссертации):

п.1. ***Модели, методы и алгоритмы проектирования и анализа*** программ и ***програмных систем, их*** эквивалентных преобразований, верификации и ***тестирования.*** [Метод организации (схема) БД ПКС с ассоциативной защитой. Метод и алгоритмы проектирования программной системы управления такими базами данных. Метод генерации тестовых БД ПКС. Анализ по результатам тестирования динамики информационных процессов в серверной части этой системы].

п.4. ***Системы управления базами данных*** и знаний. [Исследовательская версия СУБД ПКС с ассоциативной защитой].

п.8. Модели и ***методы создания программ и программных систем для параллельной*** и распределенной ***обработки данных,*** языки и инструментальные средства параллельного программирования. [Методы создания программ и программных систем моно- и мультикластерной версий серверной части СУБД ПКС с ассоциативной защитой на платформе вычислительных кластеров].

Апробация работы. Основные результаты работы неоднократно докладывались и обсуждались на: Всероссийских конференциях «Техническая кибернетика, радиоэлектроника и системы управления» (Таганрог, 2008), ДНДС-2013 (Чебоксары); Международной молодежной конференции «Туполевские чтения» (Казань, 2009, 2010, 2013); Международных конференциях «Высокопроизводительные параллельные вычисления на кластерных системах» HPC-2009, 2011-2014 (Владимир, Нижний Новгород, Пермь), «Информатика: проблемы, методология, технологии» (Воронеж, 2012), «Нигматуллинские чтения-2013» (Казань), АКТО-2014 (Казань), Инновационные технологии XXI века (Нижнекамск, 2015); Республиканском научном семинаре АН РТ «Методы моделирования» (Казань, 2012-2016), Международной конференции 2016 International Siberian Conference on Control and Communications (SIBCON-2016, Москва), Научных семинарах по информационным технологиям и защите информации (Челябинск, НИУ ЮУрГУ, 2017; Уфа, УГАТУ, 2018), Международной научно-практической конференции ИНФО-2018 (Сочи).

Публикации. Результаты диссертационной работы отражены в 20 публикациях. Среди них: 1 монография; 6 научных статей, среди которых: 3 статьи – в рецензируемых журналах из списка периодических изданий, рекомендованных ВАК (одна из них дополнительно проиндексирована в SCOPUS), и еще одна – SCOPUS-статья; 13 тезисов докладов в материалах конференций. Имеется свидетельство о государственной регистрации программы для ЭВМ.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, перечня сокращений и условных обозначений, списка терминов, списка литературы и приложений. Содержит 144 страницы машинописного текста, 52 рисунка, 8 таблиц. Список литературы содержит 119 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

ВО ВВЕДЕНИИ обосновывается актуальность темы диссертации, определяются цель и задачи исследования, приводится перечень основных результатов, выносимых на защиту. Описывается структура диссертации.

В ПЕРВОЙ ГЛАВЕ приводится аналитический обзор по имеющимся параллельным СУБД, программным решениям для работы с пространственными данными

и состоянию работ в этих областях. Дается постановка решаемой задачи с формулировкой принятых ограничений и предлагаемого подхода к ее решению.

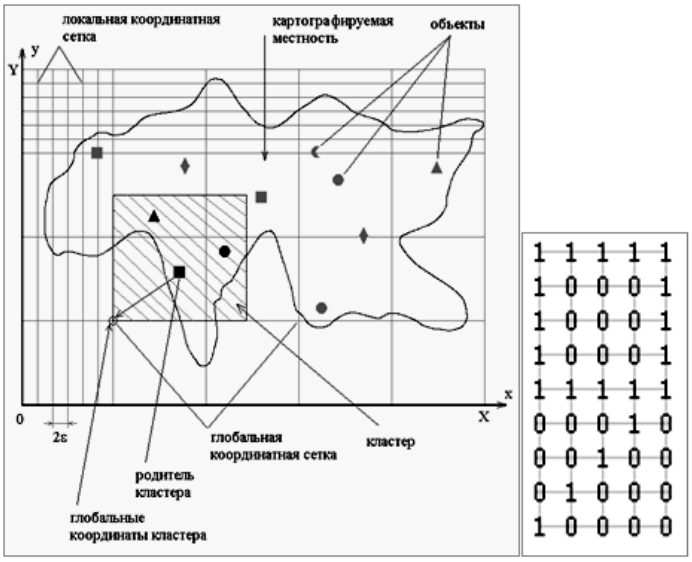
Постановка задачи

Декларативная формулировка задачи: развитие методов и алгоритмов программной организации ассоциативно-защищенных СУБД КС с целью обеспечения их полнообъектности.

Принятые ограничения.

1. *Используемое представление сцены*[2]. Для любого типа объектов используется точечное представление: точечные объекты – одна точка, линейные и площадные – последовательность точек. Исходная информация о каждом тематическом слое анализируемой сцены представлена в виде отношения с атрибутами: ИМЯ ОБЪЕКТА_ КООРДИНАТА *x*_КООРДИНАТА *y*, где *x* и *y* – координаты точки на сцене. Это отношение разбивается на множество таблиц – фрагментов. Каждый фрагмент отображает некоторый участок сцены (местности). Размеры всех участков одинаковы. Линейный размер фрагмента C=(2ε)Г >]A/Г[. Здесь A = X=Y – максимальное значение координат (x и y) картографируемого массива; Г – градационная характеристика сцены;]A/Г[– шаг глобальной координатной сетки, определяющей координаты фрагмента; ε – погрешность определения координат точек; 2ε = C/Г – шаг локальной координатной сетки внутри фрагмента. Во избежание детерминированности расположения во фрагменте его «родителя» содержимое каждого фрагмента перемешивается. Пример фрагментации сцены показан на рисунке 1а.

Используемые десятичные символы координат и кодов объектов стилизуются как бинарные изображения в виде двоичных матриц-эталонов фиксированных размеров m×n, m=2n–1 (рисунок 1б – представление символа 9 для n=5). По условию максимальное число кодов объектов равно Г.



а) б)

Рисунок 1 – а) Пример представления картографической сцены; б) Пример стилизации десятичного символа

Случайным образом генерируется набор масок для множества используемых символов. При этом знание набора масок должно быть достаточным условием правильной идентификации объектов сцены. Независимо от m, используемый алгоритм маскирования оставляет в каждой цифре от 1 до 8 значащих бит из (9n–12) существенных элементов этих матриц. Множества таких бит случайно распределены. Координаты и сами объекты кодируются 3-разрядными десятичными числами и маскируются единообразно.

Рандомизация выполняется таким образом, что без знания ключевого набора масок в каждом замаскированном объекте можно будет распознать любой из элементов используемого словаря объектов с любыми координатами на множестве случайно генерируемых наборов масок. Выполнение этого условия может быть обеспечено только при полноте множества задействованных кодов объектов и градаций их координат. При выборе n = 40-60 и использовании генератора ПСП «Вихрь Мерсенна»

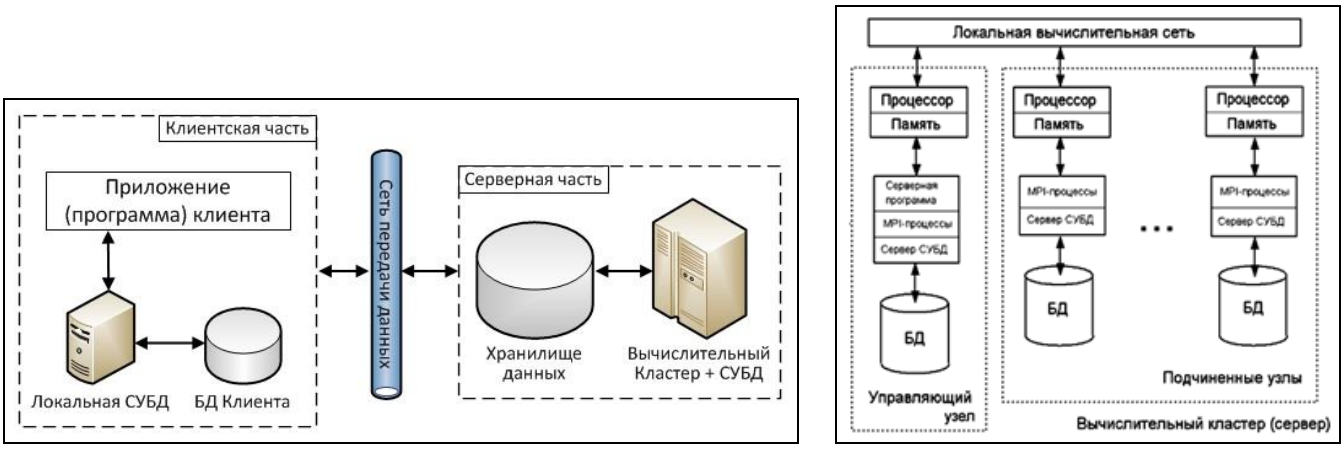
подходящая рандомизация находится с первой попытки. Мощность указанного множества $\Gamma = 10^3$. Но обычно реальное число типов объектов $\Gamma_{об} < \Gamma$. Поэтому вводятся «пустые» объекты и координаты. Число типов тех и других для случая точечных объектов $\delta = 10^3 - \Gamma_{об}$. Коды конкретного «пустого» объекта и их координат выбираются случайно на множестве из δ вариантов. Пустые объекты используются для выравнивания числа записей во всех кластерах, что затрудняет проведение стегоанализа.

2. *Принятый* АЛГОРИТМ *маскирования*[2]. Случайность сгенерированного набора масок при каждом применении АЛГОРИТМа обеспечивается случайными перестановками порядка следования эталонов цифр на разных этапах его работы и случайным выбором значащих бит эталонов. Ниже представлен полученный программным путем по АЛГОРИТМу вариант маскирования для одной из исходных перестановок десятичных цифр в случае m = 5, n = 3. Под каждой цифрой приводится соответствующая инверсная матрица маски.

0	***1***	***9***	***6***	***7***	***8***	***2***	***5***	***4***	***3***
100	100	100	100	100	100	000	000	100	100
000	000	001	000	000	000	000	000	000	001
010	000	000	000	001	010	000	000	001	000
100	000	010	100	010	100	110	110	010	010
010	010	010	010	010	010	010	010	010	010

3. *Архитектура ассоциативной СУБД КС* [2]. За основу построения специализированной СУБД ПКС с ассоциативной защитой берется «интеллектуальная» файл-серверная организация взаимодействия (рисунок 2а).

Задачи клиента: формирование пользовательских запросов с сокрытием конфиденциальных данных; отправка запросов по сети для обработки сервером; формирование размаскированной базы данных пользователя; конечная обработка (постобработка) полученных от сервера результатов. Постобработка осуществляется клиентом с помощью модуля, встроенного в ГИС.



а) б)

Рисунок 2 – а) Файл-серверная архитектура системы; б) Архитектура серверной части

Серверная часть системы, или просто Система (рисунок 2б) – это вычислительный комплекс, состоящий из управляющего узла и множества вычислительных узлов. Управляющий узел решает задачи получения запросов от клиентов, распределения задач между вычислительными узлами кластера, сбора результатов. Остальные узлы кластера являются вычислительными.

Предлагаемый подход к решению задачи. Задача решается путем разработки натурной модели Системы с использованием двух архитектурных подходов:

1. *Монокластер* («весь кластер на запрос»). В каждый момент времени все вычислительные узлы серверной части обрабатывают только один запрос. Особенности:
- Базы данных равномерно распределяются по узлам кластера.
- Каждый вычислительный узел выполняет поиск удовлетворяющих условиям запроса элементов в своей локальной части БД.
- Конечный результат получается путем конкатенации частичных результатов со всех узлов, где эти результаты были получены.
- Серверу нет необходимости выполнять функцию роутера. Запросы обрабатываются в порядке очередности их поступления.

2. *Мультикластер* («один узел на один запрос»). Параллельное выполнение множества клиентских запросов на кластере. Система способна обрабатывать одновременно до N запросов, где N – число узлов. Базы данных реплицируются по узлам. На управляющий узел возлагаются дополнительные функции распределения запросов между узлами – функция роутера.

Натурная модель состоит из двух отдельных программных модулей (по числу архитектур). Переключение между ними происходит в ручном режиме. Для этого предварительно необходимо закончить обработку очереди запросов и переформатировать БД ПКС для работы с запускаемым модулем.

Система базируется на сервере стандартной СУБД MySQL. Разрабатывается на языке программирования Си/C++ с применением библиотеки параллельного программирования MPI. Используемая платформа – вычислительный кластер Sun Blade 6048 с 22 узлами Sun x6250.

Для проведения экспериментов разработан представительский тест (ПТ) из 20 селективных запросов. Выборка объектов − по координатам. Например, запрос:

*select * from <kbd_name> where X>230000 and X<570000 and Y>1570006 and Y<718000.*

Здесь *kbd_name* – имя соответствующей БД КС (точечных, линейных или площадных объектов).

Сравнительная оценка производительности обеих архитектур проводится по двум показателям: 1) суммарное время обработки пакета запросов $T_{общ}$, 2) время задержки $T_{зд}$ для каждого запроса от момента его поступления в систему до момента отправки ответа $T_{зд} = T_{обр} + T_{ожд} + T_{пробр}$. Здесь: $T_{обр}$ – время обработки запроса сервером, $T_{ожд}$ – время простоя запроса в очереди запросов сервера, $T_{пробр}$ – время предобработки запроса, включающее раскрытие сокрытых в запросе параметров, синтаксический, семантический анализ, анализ прав пользователя на исполнение запроса и т.п.

Полученные данные используются для подсчета математического ожидания $M(T_{зд}) = (\sum_{1}^{n} T_{зд}^{k})/n$ и среднеквадратического отклонения $\sigma(T_{зд}) = \sqrt{M[T_{зд} - M(T_{зд})]^2}$, k – номер запроса, n – число обработанных запросов.

Окончательное решение задачи связывается с выработкой по результатам эксперимента практических рекомендаций по реализации серверной части СУБД ПКС с ассоциативной защитой и разработкой программной системы клиентской части.

ВО ВТОРОЙ ГЛАВЕ предлагается метод организации (схема) БД ПКС с ассоциативной защитой. Обсуждаются: особенности фрагментации, перемешивания и выравнивания размеров фрагментов, используемых при формировании БД ПКС, возможные типы запросов к БД ПКС. Приводятся: метод стохастической генерации тестовых баз данных, алгоритмы и программная реализация их формирования на кластере, метод обработки пользовательских запросов без полного раскрытия базы данных.

Предлагаемая инфологическая схема **БД ПКС** представлена на рисунке 3. Она учитывает специфику представления картографической сцены и необходимость хранения данных в сокрытом виде. Схема предполагает единообразный формат хранения данных для всех типов объектов: точечных, линейных и площадных.

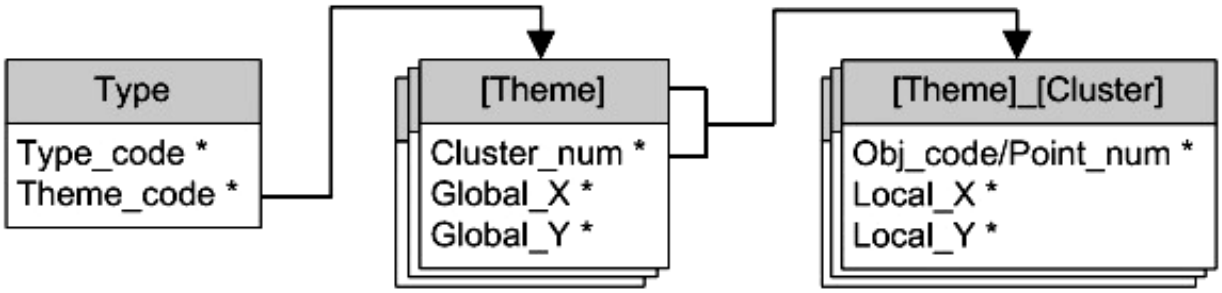


Рисунок 3 – Предлагаемая инфологическая схема БД ПКС

Особенностью схемы является выделение наборов таблиц – отдельная таблица для каждого тематического слоя (описываются фрагменты-кластеры внутри слоя), и для каждого фрагмента (описываются объекты внутри него). Взаимодействие между табличными отношениями определяется процедурами программы-сервера в ходе обработки запроса. Связи на рисунке 3 означают формирование SQL-запроса на поиск в таблице, имя которой находится после раскрытия стегокода из предыдущей таблицы.

Метод организации БД ПКС согласно данной схеме:

- Десятичные коды типов слоев и коды самих тематических слоев задаются при генерации БД. Коды выбираются случайным образом из всего множества.
- Коды тематических слоев для каждого типа уникальны.
- Узловые точки линейных и площадных объектов нумеруются от *начала объекта к его концу*. Нумерация определяется очередностью записей об узловых точках в исходном файле.
- Пустые точечные объекты и пустые узловые точки каждого отношения имеют номера, превышающие число записей.
- Число записей для всех отношений одинаково.
- Порядок записей в отношениях типа *[Theme]_[Cluster]* случаен.

Описание отношений:

- *Type* – отношение, содержащее информацию о всех тематических слоях сцены, представленных парой сокрытых кодов: Код типа – Код слоя.
- *[Theme]* – набор отношений, каждое из которых описывает отдельный тематический слой (кластеры внутри слоя). Название отношения есть открытый код этого слоя.
- *[Theme]_[Cluster]* – набор отношений, каждое из которых описывает содержимое (объекты) одного кластера какого-либо тематического слоя. Название отношения составное, содержит открытый код слоя и код кластера.
- *Type_code** определяет тип слоя: слой точечных, слой линейных или слой площадных объектов. Используется при работе программы.
- *Theme_Code** – уникальный код тематического слоя.
- *Cluster_num** – код номера фрагмента в тематическом слое.
- *Global_X*, Global_Y** – глобальные координаты нижнего левого угла данного фрагмента.
- *Obj_code / Point_num ** – код точечного объекта (код номера узловой точки линейного или площадного объекта).
- *Local_X*, Local_Y** – локальные координаты точки/узла внутри фрагмента.
- Данные по атрибутам, отмеченным звездочкой, хранятся в *замаскированном виде*.

Семантическая информация картографических объектов хранится у клиента в отдельной открытой БД. К этой информации клиент может обратиться, используя в качестве идентификатора полученный от сервера код объекта.

*Формирование фрагментов***.** Случайным образом выбирается точечный объект или узловая точка линейного/площадного объекта, не содержащийся ни в одном из уже сформированных фрагментов сцены. На основе этой точки создается фрагмент сцены с соответствующим порядковым номером, координаты левого нижнего угла которого совпадают с координатами левого угла ячейки глобальной координатной сетки, где расположен выбранный объект. Код порядкового номера фрагмента *i*, координаты фрагмента (левый нижний угол) заносятся в рабочее отношение [Наименование слоя], описывающее фрагменты данного слоя. Все непокрытые ранее созданными фрагментами точки входят в новый фрагмент. Если на карте имеются точечные объекты или узловые точки линейных/площадных объектов, не вошедшие ни в один из сформированных фрагментов, то переходим к началу. Иначе завершаем алгоритм.

*Перемешивание***.** Порядок следования строк в табличных отношениях, описывающих информацию об объектах внутри фрагментов картографической сцены – коды точечных объектов или номера узловых точек линейных/площадных объектов и их локальные координаты, для существенных и несущественных узловых точек и объектов делается случайным.

Добавление пустых точечных объектов и узловых точек. Коды пустых точечных объектов данного тематического слоя выбираются случайным образом на множестве кодов объектов соответствующего типа, не задействованных в этом слое. Для этого определяется число точечных объектов N^{points}_{max} в самом насыщенном этими объектами фрагменте в рамках данного тематического слоя. Каждый фрагмент в указанном слое дополняется пустыми точечными объектами до величины N^{points}_{max}. Аналогично для несущественных узловых точек линейных и площадных объектов.

Отсеивание пустых точечных объектов и узловых точек линейных/площадных объектов при работе с базой данных клиента происходит с помощью дополнительного отношения Empty_nums. Упрощенный формат записи кортежей в этом отношении показан ниже. Записи в данном отношении хранятся в замаскированном виде.

theme_type	theme_code	N_{max}

Наличие взаимосвязанной нумерации узловых точек линейных и площадных объектов не накладывает дополнительных ограничений на линейные размеры таких объектов и на размеры фрагментов, генерируемых при формировании сцены.

Возможные типы запросов к БД ПКС:

1. Одноуровневые *селективные* запросы без вложенностей.

Запросы от пользователей приходят на файл-сервер в следующем виде:

*select * from <имя_БД> [where {x,y} >=,<=,>,<,= <координата>];*

Условие *where* используется для установления координатных границ области поиска в рамках сцены. В качестве координаты – физические координаты исходной картографической сцены. Отсутствие условия приведет к выборке всех объектов из указанной БД КС. Имя базы данных заменяется на необходимые имена табличных отношений.

2. Запросы *изменения*:

- *Добавление* объектов (операция insert). Кроме очереди входящих запросов, на сервере имеется табличное отношение – буфер для приема от клиентов информации о добавляемых точечных объектах или узловых точках линейных/площадных объектов. Атрибуты этого отношения*: Query_num, Point_num, Coord_X, Coord_Y, Semantic information.*

- *Удаление* объектов (операция delete). Формат запроса удаления (*delete*) аналогичен формату селективного запроса. Точечные объекты, попавшие в указанную координатную область (условие *where*) удаляются полностью из всех тематических слоев. В случае линейных или площадных объектов возможны два подхода:

– Объект удаляется из БД целиком, даже если не все его узловые точки попали в область условия.

– Удаляются лишь узловые точки объекта, попавшие в область условия. При этом форма объекта может существенно измениться – более трудоемкий подход.

- *Обновление* записей об объектах (операция update). Реализовано как совокупность двух вышеописанных операций.

Метод формирования тестовых БД ПКС. Для проведения экспериментов, на множестве запросов представительского теста (ПТ) были сгенерированы в режиме «монокластер» тестовые БД ПКС по следующему методу:

1. Любой запрос ПТ формирует некоторую прямоугольную область поиска, ограниченную указанными в запросе координатами *X* и *Y*. Выбираются все объекты, удовлетворяющие условию. Области, отвечающие всем запросам ПТ, не пересекаются, их размеры различны.

2. Генерируются по три БД КС разных размеров для каждого типа объектов (всего 9) на области размером 2000 км × 2000 км. Отношения объемов баз данных объектов одного типа – БД1: БД2 : БД3 ≈ 1 : 2 : 4. Числа объектов и узловых точек в генерируемых БД одного типа находятся в тех же отношениях. Размещение объектов на плоскости при генерации БД случайно, но строго отвечает условиям покоординатной выборки для каждого из 20 запросов ПТ. На рисунке 4 показан пример размещения линейных объектов на участке местности. Каждому прямоугольнику соответствует один селективный запрос. Объекты находятся только внутри указанных прямоугольников.

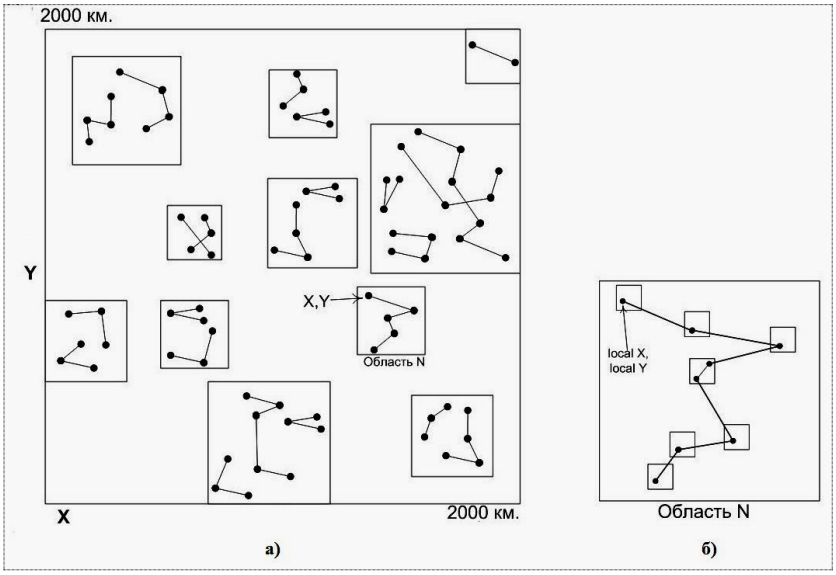


Рисунок 4 – Подход к формированию сцены тестовой БД

3. Координаты для точечных объектов и узловых точек линейных и площадных объектов присваиваются на интервале [0, 1999999] в результате стохастической генерации. После получения БД КС в открытом виде, выполняется маскирование всех полей в каждом табличном отношении БД КС. Для этого применяется библиотека расширения функционала СУБД MySQL *universal_udf*. Библиотека содержит в себе две дополнительных функции: *udf_decipher* (раскрытие данных) и *udf_cipher* (маскирование полей таблиц).

Пример программного кода:

```
conn = mysql_init (NULL);
mysql_real_connect (conn,"localhost","root",NULL,"DB_NAME", 3306, NULL, 0)
sprintf(query,"update %s set local_x=udf_cipher(local_x),
local_y=udf_cipher(local_y);",row[0]);
mysql_query (conn,query);
```

Здесь: *DB_NAME* – имя картографической базы данных, информация в которой будет подвержена сокрытию. Выражение local_x=*udf_cipher*(local_x) указывает СУБД заменить поле local_x (открытые данные) определенной таблицы на результат выполнения операции маскирования *udf_cipher* (local_x).

Метод обработки пользовательских запросов без полного раскрытия базы данных. Суть предлагаемого метода состоит в следующем.

1. Согласно условию селективного запроса выбирается множество тематических слоев данного типа. «Сокрытые» коды *Type* и *Theme* таблицы **Type** предварительно раскрываются вплоть до получения нужного открытого кода *Theme*.

2. В каждом таком слое выбираются все фрагменты, глобальные координаты которых удовлетворяют условию запроса. Их номера запоминаются в таблице вида:

Код слоя	№ кластера
Например, 005	Например: 035, 047, …

3. В каждом найденном фрагменте по его номеру раскрываются поля координат всех записей, но отбираются и сохраняются лишь те записи, которые удовлетворяют условию запроса. При поиске выполняется преобразование координат. В итоге получаем таблицу:

Код слоя	Код объекта (№ узл. точки)	Координата X	Координата Y
Например, 005	Например, 135		

4. Полное сокрытие всех полей сформированной таблицы и ее отсылка клиенту по открытому каналу связи.

В качестве примера рассмотрим пользовательский запрос на выборку линейного объекта с кодом 125 с заданными координатными условиями. При формировании запроса указывается код типа слоя: 284 – тематические слои линейных объектов.

select * from kbd1.284 where X<20000 and Y>50000 and code=125;

Согласно рассмотренным пунктам метода в процессе обработки этого запроса генерируются следующие подзапросы:

1. **SELECT Theme_code from Type where Theme_code= 125 and Type_code=284;**

Результатом данного запроса станет получение кодов тематических слоев, в рамках которых ведется поиск. Если имя слоя не указано, то работа ведется со всеми имеющимися тематическими слоями, перечисленными в отношении Line_Themes. Кроме поиска по имени, тематические слои могут выбираться по номерам кода. В таком случае добавляется условие вида: **udf_decipher(Code) > 35**. Если результатом поиска станет пустое множество, то пользователь получит соответствующее сообщение.

2. **SELECT udf_decipher Cluster_num from 125 where udf_decipher (Global_X)·A_1 < 20000 and udf_decipher (Global_Y)·A_1 > 50000 – A_1;**

Результатом запроса станет список фрагментов слоя N. Здесь **Lines.N** – табличное отношение, описывающее слой N, где N = Code (результат подзапроса1). При наличии в исходном запросе пользователя координатных условий, на сцене выбираются кластеры, отвечающие этим условиям. Переменная $\mathbf{A_1}$ – шаг глобальной координатной сетки на сцене.

3. Для каждого из выбранных в подзапросе 2 кластеров:

SELECT udf_decipher (Point_num), udf_decipher (local_X) + udf_decipher(Global_X)·A_1, udf_decipher(local_Y) + udf_decipher (Global_Y)·A_1 from 125, 125_i where (udf_decipher (local_X) + udf_decipher (Global_X)·A_1) < 20000 and (udf_decipher (local_Y) + udf_decipher (Global_Y)·A_1) > 50000;

Здесь: ***i*** – порядковый номер выбранного кластера;

udf_decipher (local_X) + udf_decipher(Global_X)·A_1 – физическая координата объекта на сцене, полученная в результате обратного преобразования координат.

Итогом станет подмножество ненулевых результатов, содержащее описание узловых точек линейных объектов в формате *Код слоя – Номер точки – Координата X – Координата Y*. В обработке запроса первоочередную роль выполняет пользовательская функция расширения MySQL udf_decipher(), «читающая» содержимое стегоконтейнеров без их раскрытия. Временные файлы или отношения при этом не создаются.

В ТРЕТЬЕЙ ГЛАВЕ дается метод организации серверной части исследовательского прототипа системы Map Cluster для обеих архитектур: моно- и мультикластерной. Описываются алгоритмы обработки замаскированной информации, взаимодействие программных модулей. Приводятся результаты выполненных экспериментальных исследований с анализом этих результатов.

Функции программных блоков Системы. Для архитектуры *«монокластер»:* Блок *Analyze* – считывание запроса из очереди, проверка прав пользователя, раскрытие сокрытых параметров запроса, семантический/синтаксический анализ запроса. Блок параллельной обработки *Process* – инициализация параллельной обработки, тиражирование запроса по вычислительным узлам кластера, выполнение запроса каждым вычислительным узлом, сбор промежуточных данных на узлах, сбор данных управляющим узлом с вычислителей. Блок формирования результата *Result* – формирование результата на управляющем узле; отправка результатов клиентам по сети. В системе с архитектурой «мультикластер», в дополнение к вышеуказанным, введены программные блоки-модули: *Route* – на управляющем узле; *Process* – на каждом вычислительном узле.

Выполнение запросов. Формат запроса *удаления* (*delete*) аналогичен формату селективного запроса. Точечные объекты, попавшие в указанную координатную область (условие *where*) удаляются полностью из всех тематических слоев. Условием для поиска и последующего удаления могут быть координаты и/или коды объектов.

Например, результатом выполнения запроса

*delete * from points where X > 23000 and Y > 37500;*

должно стать удаление из соответствующей базы данных всех точечных объектов, координаты *X* и *Y* которых больше 23000 и 37500 метров соответственно.

Блок-схема программного алгоритма для операции удаления аналогична блок-схеме для операции селекции за небольшим отличием:

– В блоке параллельной обработки на вычислительных узлах не собираются промежуточные данные. Вместо этого фиксируется результат выполнения запроса.

– Управляющий узел отправляет клиенту информацию об успешности выполнения запроса или код ошибки в случае отрицательного результата.

Экспериментальные исследования были ограничены случаем селективных запросов. Причина в том, что при работе мультикластера необходимо следить за *когерентностью* данных. Если в систему приходит запрос на изменение хранимых данных, такой запрос должен выполняться всеми узлами по завершении обработки в них текущих запросов. Это приводит к появлению больших пиков на графике времени задержек. В монокластере такого не наблюдается.

Проводились три эксперимента для обеих архитектур системы:

1. Установление характера изменения коэффициента ускорения системы для архитектуры монокластера с ростом числа узлов N на ПТ из 20 запросов на всех сгенерированных ПБД КС для каждого типа объектов. Ускорение системы $S = T_{общ\ 1}/T_{общ\ N}$.

2. То же для архитектуры мультикластера на тех же ПТ.

3. Сравнение величин $M(T_{зд})$, $T_{общ}$ обеих архитектур на непрерывном потоке из 1000 запросов.

В ходе экспериментов 1 и 2 было установлено, что поведение графиков **S** для всех трех типов объектов аналогично. Поэтому на рисунках 5 и 6 приводятся результаты только для БД точечных объектов. По оси абсцисс – число узлов в кластере. По оси ординат – коэффициент ускорения системы.

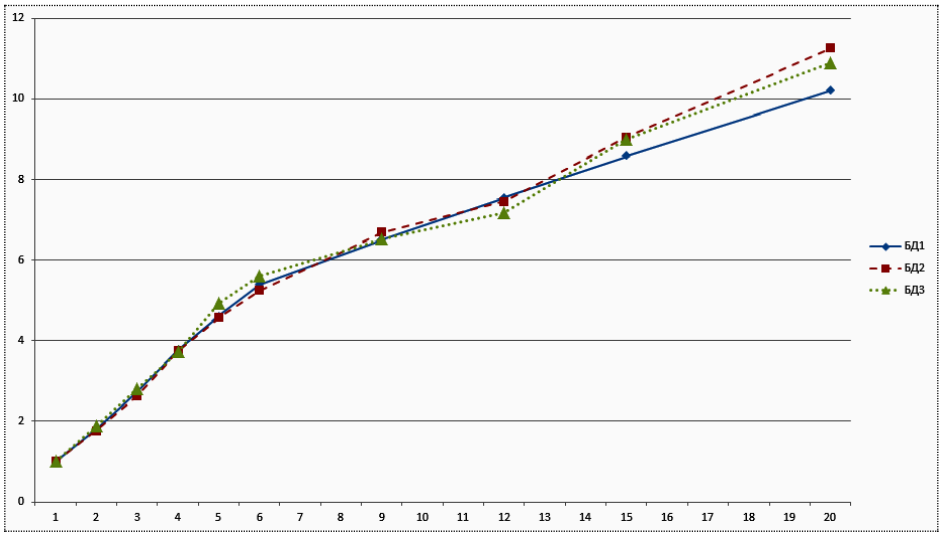


Рисунок 5 – Ускорение монокластера при обработке ПТ

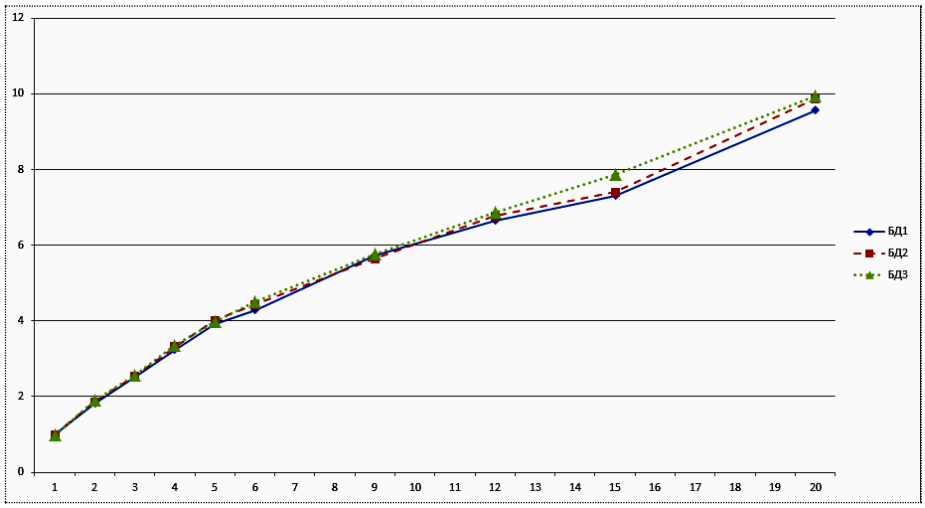


Рисунок 6 – Ускорение мультикластера при обработке ПТ

Замечаем, что монокластер ускоряется сильнее мультикластера при любом числе узлов N. Особенно это заметно при малых N (меньше 6) и при N_{max}=20 со средней и большой БД ПКС. Монокластер превосходит мультикластер и по масштабируемости. Принимая за границу масштабируемости число уздов $N_{гр}$, при котором показатель эффективности Э=$S/N_{гр}$= 1/2,согласно рисункам 5,6 для условий эксперимента получаем $N_{гр}^{моно} > N_{гр}^{мульти} = 20$.

Эксперимент 3. Сравнение проводилось на БД точечных объектов среднего размера – БД2, при N=4 и N=9 для обеих архитектур. В таблице 1 приведены полученные

значения M ($T_{зд}$), $T_{общ}$ для потока из 1000 запросов, случайно сформированного на множестве *78 запросов, не обязательно отвечающих оговоренному ранее условию формирования ПТ (более реальный сценарий).*

Таблица 1 – Результаты обработки непрерывного пакета запросов

	Среднее время задержки $M(T_{зд})$, сек	$\sigma(T_{зд})$, сек	Время обработки пакета $T_{общ}$, сек
Монокластер, 4 узла	122	45,8	10610
Мультикластер, 4 узла	76	29,7	8172
Монокластер, 9 узлов	55	21,5	4875
Мультикластер, 9 узлов	37	14,2	4152

При N=4 мультикластер имеет существенный выигрыш по $M(T_{зд})$ ~ на 38%, величина $T_{общ}$ меньше ~ на 23%, а $\sigma(T_{зд})$ ~ на 35% . При N=9 мультикластерная архитектура по-прежнему лучше по значениям этих параметров: $M(T_{зд})$ ~ на 32%, $T_{общ}$ ~ на 17%, $\sigma(T_{зд})$ ~ на 33%.

Для выявления динамики поведения кластера с ростом числа узлов на разных этапах его функционирования был выполнен дополнительный эксперимент (*эксперимент 4)* для монокластера. В ходе эксперимента выявлялось общее время выполнения каждого этапа при обработке пакета пользовательских запросов. За суммарный объем работ на каждом этапе принято суммарное число эквивалентных скалярных операций длительностью 1 сек. каждая.

При обработке запроса выделяются 8 основных этапов: *Этап 1* - Ожидание запроса в буфере запросов; *Этап 2* - Считывание запроса из очереди, обработка сокрытых параметров; *Этап 3* - Тиражирование запроса по вычислительным узлам; *Этап 4* - Поиск объектов в БД на вычислительных узлах согласно запросу; *Этап 5* - Формирование промежуточного результата на вычислителях для отправки на управляющий узел; *Этап 6* - Сбор результатов с вычислительных узлов; *Этап 7* - Формирование конечного результата; *Этап 8* - Отправка результатов пользователю.

Полученные гистограммы, кроме этапа 1, отображены на рисунке 7.

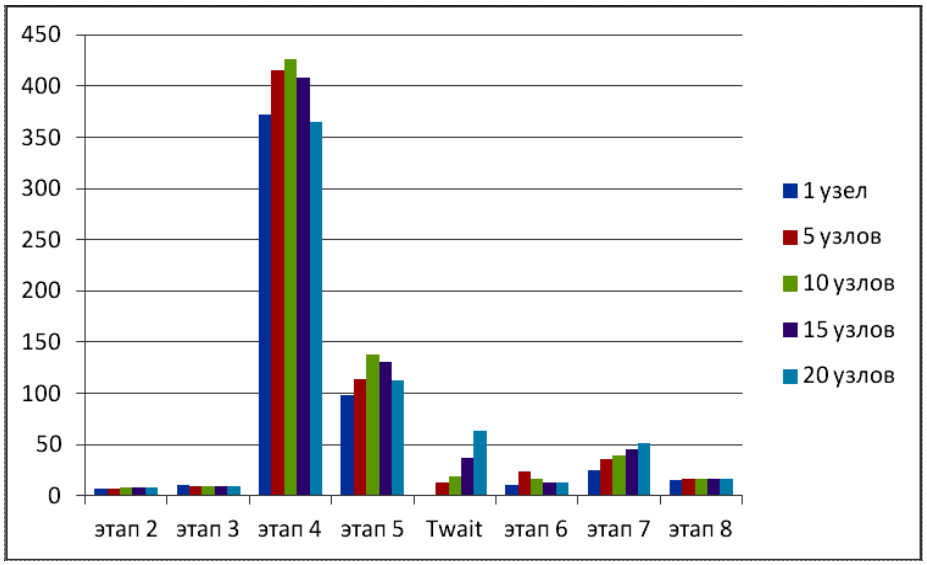


Рисунок 7 – Суммарное время выполнения отдельных этапов запросов в монокластере

Обсуждение результатов экспериментов. Результаты, полученные в экспериментах 1 и 2, показывают преимущества монокластера по значениям параметров S и $N_{гр}$. Эксперимент 3 отдает предпочтение мультикластеру. Однако сценарий, использованный в этом эксперименте, видится маловероятным для небольшого числа

пользователей. Мультикластерный подход существенно проигрывает монокластеру при выполнении запросов изменения (добавление, удаление) базы данных и при выполнении селективных запросов с выборкой крупных областей со множеством объектов. Выбор монокластера однозначен при генерации новых БД ПКС и выполнении операции смены ключа для существующих БД ПКС. Поэтому наш главный вывод состоит в том, что *архитектура монокластера предпочтительна для практического использования.*

Из рисунка 7 видно, что с ростом числа узлов в монокластере начинает существенно нарастать суммарное время простоя исполнительных узлов T_{wait}. Это время обуславливается наличием в системе барьерной синхронизации между исполнительными узлами. Возможный путь ускорения системы в архитектуре «монокластер» – переход к асинхронной обработке запросов и совмещение процедур, выполняемых параллельно на исполнительных узлах, с некоторыми этапами на управляющем узле.

В ЧЕТВЕРТОЙ ГЛАВЕ описывается подход к построению клиентской части системы. Предлагаемая схема локальной базы данных клиента показана на рисунке 8.

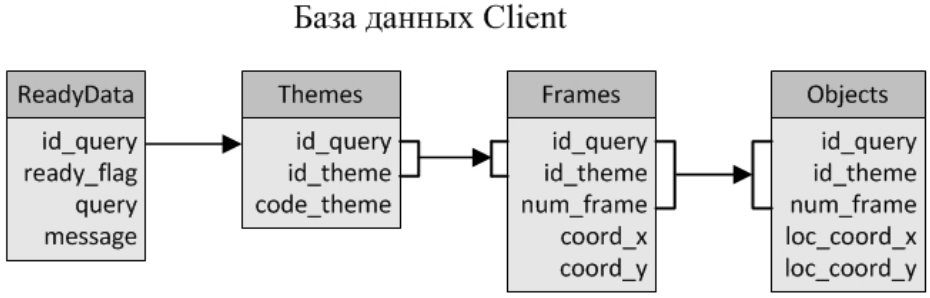


Рисунок 8 – Схема БД клиента

Здесь *ReadyData* – таблица хранения запросов, формируемых пользователем (буфер). Для запроса определяется идентификационный номер (поле *id_ query*), флаг готовности результата *ready_ flag* устанавливается в 0. В группу связанных таблиц *Themes, Frames, Objects* записываются результаты обработки запроса серверной частью. Эти таблицы связываются между собой и с таблицей *ReadyData* при помощи ключевого поля *id_query*. Информация передается клиенту в сокрытом виде.

Программа пользователя формирует запросы пользователя с проверкой прав доступа, синтаксическим анализом запроса, сокрытием конфиденциальных параметров запроса и отправкой запроса на сервер; организует получение обработанной на сервере информации с переводом ее в открытый вид и сокрытием следов обработки. Связь клиентских узлов и сервера происходит при помощи интерфейса сетевых соединений MySQL C API.

Запросы пользователей и данные по объектам сцены, являющиеся результатом запроса, замаскированы и передаются между сервером и клиентом по открытым каналам связи. Для работы с системой клиенту надо иметь имя пользователя, пароль для подключений и стегоключ.

Разработан алгоритм формирования контуров линейных и площадных объектов на основе данных, полученных от серверной части СУБД ПКС в ответ на поставленные пользователем запросы.

В ЗАКЛЮЧЕНИИ представлены основные выводы по работе.

ОСНОВНЫЕ НАУЧНЫЕ РЕЗУЛЬТАТЫ

1. Предложен метод организации (схема) БД ПКС с ассоциативной защитой (АЗ), основанный на разделении тематических слоев для разных типов объектов с выде-

лением своего слоя для любого линейного и площадного объекта, что, в отличие от известного, позволяет снять ограничения на размеры протяженных объектов.

2. Предложен метод генерации тестовых БД ПКС с АЗ, основанный на выделении в сцене прямоугольной области для каждого запроса представительского теста, что, в отличие от универсальных тестов TPC, позволяет провести анализ динамики СУБД ПКС при обработке пакетов запросов.

3. Предложен метод обработки селективных запросов к БД ПКС, основанный на неполной выборке узловых точек протяженных объектов, что, в отличие от предложенного ранее подхода, позволяет провести обработку таких запросов по указанным объектам без «раскрытия» всей БД. На основе метода разработан программный комплекс (натурная модель, воспроизводящая реальные системные ситуации) серверной части ассоциативно-защищенной СУБД ПКС с ассоциативной защитой, который позволяет проводить исследования по ассоциативной стеганографии с использованием адекватного инструментального средства.

4. На разработанном программном комплексе проведен ряд экспериментальных исследований, что позволило получить сравнительные характеристики динамики процессов при обработке пакетов запросов для двух возможных архитектур системы и на их основе предложить практические рекомендации по применению архитектур системы и возможному улучшению производительности.

5. Предложен программный алгоритм клиентской части системы, основанный на предварительной обработке запросов пользователя, что в отличие от существующих подходов позволяет пользователю выполнять запросы без знания структур БД ПКС. На основе алгоритма разработан программный прототип клиентской части системы.

Перспективы дальнейшей разработки темы. В рамках дальнейших исследований планируется развить разработанные методы и адаптировать для работы на одноузловых многоядерных системах с использованием графических ускорителей. Также отдельного изучения потребуют вопросы ускорения выполнения запросов изменения в мультикластере, вопросы балансировки нагрузки в системе, а также методы распределения БД ПКС по узлам монокластера.

СПИСОК ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

В рецензируемых журналах из перечня ВАК

1. Raikhlin, V.A. Reliable Recognition of Masked Binary Matrices. Connection to Information Security in Map Systems /V.A. Raikhlin, I.S. Vershinin, R.F. Gibadullin, S.V. Pystogov //Lobachevskii Journal of Mathematics, 2013, Vol. 34, No. 4, pp. 319–325. – ***Scopus***.

2. Пыстогов, С.В. Сравнение архитектурных решений для полнообъектной СУБД картографических сцен с ассоциативной защитой /Пыстогов С.В. //Вестник КГТУ им. А.Н.Туполева. 2014. №3. С.64-79

3. Вершинин, И.С. Импорт/экспорт ассоциативно защищенных картографических данных с их обработкой в системе Security Map Cluster /И.С. Вершинин, Р.Ф. Гибадуллин, С.В. Пыстогов, М.Ю. Перухин //Вестник Казан. технологического университета, 2015. – № 10. – С. 174-180.

Монография

4. Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В., Райхлин В.А. Ассоциативная стеганография (Приложение к анализу сцен) /Под ред. В.А. Райхлина – Казань: Изд-во Казан. ун-та, 2014. – 132 с. ISBN 978-5-00019-284-9

Другие публикации

5. Гибадуллин Р.Ф., Прохоров А.Е., Пыстогов С.В. Управление защищенными картографическими базами данных на вычислительном кластере //Техническая кибернетика, радиоэлектроника и системы управления: Материалы 9-й Всерос. науч. конф. – Таганрог: ТТИ ЮФУ, 2008. – С. 102-103.

6. Гибадуллин Р.Ф., Пыстогов С.В. Обработка зашифрованных данных посредством СУБД MySQL //Туполевские чтения: Материалы 17-й Междунар. молод. научн. конф. – Казань: КГТУ, 2009. Т.4. С.60-62.

7. Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В. Использование кластерных технологий при решении задач защиты картографических данных //Высокопроизводительные параллельные вычисления на кластерных системах (HPC-2009): Материалы 9-й Междунар. конф. – Владимир: ВлГУ, 2009. – С. 68-72.

8. Гибадуллин Р.Ф., Пыстогов С.В. Клиентская часть системы управления защищенными картографическими базами данных //Туполевские чтения: Материалы 18-й Междунар. молод. научн. конф. – Казань: КГТУ, 2010. Т.4. С.97-99

9. Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В. Параллельная СУБД с ассоциативной защитой картографических данных //Высокопроизводительные параллельные вычисления на кластерных системах (HPC-2011): Материалы 11-й Всерос. конф. – Нижний Новгород: ННГУ, 2011. – С. 46-49.

10. Гибадуллин Р.Ф., Пыстогов С.В. Параллельная система управления полнообъектными защищенными базами данных картографических сцен //Высокопроизводительные параллельные вычисления на кластерных системах (HPC-2012): Материалы 12-й Всерос. конф. – Нижний Новгород: ННГУ, 2012. – С. 91-95.

11.Пыстогов, С.В. Параллельная СУБД стегозащиты картографических данных //Информатика: проблемы, методология, технологии: материалы XII Международной научно-методической конференции – Воронеж, 2012. – С.341-342.

12. Гибадуллин Р.Ф., Пыстогов С.В. Формирование защищенных картографических баз данных с учетом погрешности локализации объектов в системе Security Map Cluster //Туполевские чтения: Материалы 20-й Междунар. молод. научн. конф. – Казань: КГТУ, 2013. – Т.3, Ч.1. – С. 97-99.

13. Гибадуллин Р.Ф., Пыстогов С.В. Подходы к организации системы управления защищенными картографическими базами данных //Высокопроизводительные параллельные вычисления на кластерных системах (HPC -2013): Материалы 13-й Всерос. конф. – Нижний Новгород: ННГУ, 2013. – С. 77-81.

14. Райхлин В.А., Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В. Двумерно-ассоциативная защита информации в картографических системах //«Нигматуллинские чтения-2013»: Материалы междунар. научно-технич.конф., 19-21 ноября 2013 г.– Казань: Изд-во Казан. гос. техн. ун-та, 2013. – С. 48-50.

15.Пыстогов, С.В. Моделирование процессов в файл-сервере СУБД полнообъектных картографичских сцен с ассоциативной защитой //Моделирование систем: Труды Республиканского научного семинара «Методы моделирования».− Казань: «ФЭН» (Наука), 2013. Вып.5. С.100-110.

16. Пыстогов, С.В. Сравнение двух архитектур серверной части системы управления защищенными базами данных картографических сцен //АКТО-2014: Материалы междунар. научно-практ. конф. – Казань: Изд-во Казан. Гос. Техн.ун-та, 2014. С. 414-418.

17. Гибадуллин Р.Ф., Пыстогов С.В., Вершинин И.С. Сервер и клиент систем управления картографическими базами данных //Инновационные технологии XXI века: материалы Междунар. научно-практ. конф. – Казань: Изд-во Казан. гос. техн. ун-та, 2015. – С. 5-8.

18. Raikhlin V.A., Vershinin I.S., Gibadullin R.F., Pystogov S.V. Reliable Recognition of Masked Cartographic Scenes During Transmission over the Network //2016 International Siberian Conference on Control and Communications (SIBCON-2016). – ***Scopus.***

19. Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В., Райхлин В.А. Анализ ассоциативно защищенных картографических сцен //Труды Республиканского научного семинара «Методы моделирования». – Казань: «ФЭН» («Наука») АН РТ, 2016. – Вып. 6. – С. 94-116.

20. Пыстогов С.В. Анализ исследования специализированной параллельной системы управления маскированными картографическими базами данных // Инновационные, информационные и коммуникационные технологии (ИНФО-2018): Сборник трудов XIV Международной научно-практической конференции / под.ред. С.У.Увайсов – Москва: Ассоциация выпускников и сотрудников ВВИА им. проф. Жуковского, 2018. – С.113-118.

Свидетельство о государственной регистрации программы СУБД ПКС

21.Вершинин И.С., Гибадуллин Р.Ф., Пыстогов С.В. Программа управления ассоциативно защищенными картографическими базами данных «Security Map Cluster» // Свидетельство о государственной регистрации программы для ЭВМ №2016611421.

Диссертант С.В.Пыстогов