МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по выполнению лабораторных работ
по дисциплине
«Математические методы представления данных»
для студентов направления подготовки
02.03.01 Математика и компьютерные науки
направленность (профиль)

«Анализ данных и искусственный интеллект»

Оглавление

Лабораторная работа№1. Введение в системы распределенного хранения и обработки данных

Цель работы: овладение программными инструментами решения простейших вычислительных задач, возникающих в процессе проектирования распределенных систем хранения данных.

Формула для выбора варианта:

Вариант = ((номер по списку) mod (количество вариантов)) + 1.

Задание 1. Разработать программу для перевода неотрицательных чисел из десятичной в двоичную систему счисления. Программа должна учитывать возможность перевода чисел, содержащих дробную часть.

Входные данные:тестовое число в десятичной системе счисления.

Выходные данные: число в двоичной системе счисления.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 2. Разработать программу для генерации сочетаний и перестановок из *n*по *k*различных элементов и вычисления мощности полученных множеств.

Входные данные:*n*; *k*.

Выходные данные: множество сочетаний; множество перестановок; мощность сгенерированных множеств.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 3. Решить задачу. Компьютерная кластерная система, решающая некоторую сложную вычислительную задачу, разбита на *sub_grid* подсистем. Каждая подсистема решает изолированнуювычислительную подзадачу. Для решения общей задачи необходимо решить все подзадачи. Каждая из подсистем является объединением n_iвычислительных узлов, $i = 1..sub\_grid$.Для корректного выполнения подзадачи необходимо, чтобы нормально функционировали k_i из n_i вычислительных узлов подсистемы,$k_i < n_i$, $i = 1..sub\_grid$. Подсчитать вероятность *P*корректного решения вычислительной задачи, если вычислительные узлы в подсистемах однотипные и нормально функционируют с вероятностью p_i, $i = 1..sub\_grid$.

Входные данные:

sub_grid – целое число из диапазона [3, 5],генерируется случайно;

n_i – целые числа из диапазона [20, 30],генерируются случайно, $i = 1..sub\_grid$;

k_i – целые числа из диапазона [10, n_i – 5],генерируются случайно, $i = 1..sub\_grid$;

p_i –числа из диапазона [0.85, 0.95],генерируется случайно (три знака после точки), $i = 1..sub\_grid$.

Выходные данные: P.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 4. Решить СЛАУ $AX = B$.Размерность матриц A – $n \times n$, B – $n \times 1$.

Входные данные:

n – целое число из диапазона [10, 15],генерируется случайно;

a_{ij} – коэффициенты матрицы A, целые числа из диапазона [0, 100],генерируются случайно, $i, j = 1..n$;

b_i – коэффициенты матрицы B, целые числа из диапазона [0, 100],генерируется случайно, $i = 1..n$.

Выходные данные: X.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 5. Решитьсистемусравнений, согласно варианту.

Вариант 1

$$\begin{cases} 1677x_1 - 1612x_2 + 202x_3 \equiv 1279920 \pmod{896017131}; \\ 1213x_1 - 1724x_2 + 954x_3 \equiv 50060324 \pmod{96697875}; \\ 71x_1 + 430x_2 + 95x_3 \equiv 412999872 \pmod{965066159}. \end{cases}$$

Вариант 2

$$\begin{cases} 365x_1 - 619x_2 - 1895x_3 \equiv 208549923 \pmod{238793838} \\ -478x_1 + 823x_2 + 528x_3 \equiv 179056090 \pmod{40905721(} \\ 910x_1 + 596x_2 + 1884x_3 \equiv 10236972 \pmod{88289131}. \end{cases}$$

Вариант 3

$$\begin{cases} 1465x_1 + 1875x_2 - 62x_3 \equiv 142698245 \pmod{556375666} \\ 473x_1 - 64x_2 + 1111x_3 \equiv 2808303 \pmod{18386412}; \\ -463x_1 - 1615x_2 - 389x_3 \equiv 723798 \pmod{531228236}. \end{cases}$$

Вариант 4

$$\begin{cases} 264x_1 - 125x_2 - 839x_3 \equiv 397733125 \pmod{523735558}; \\ 104x_1 - 1521x_2 + 920x_3 \equiv 8053400 \pmod{14672286}; \\ -172x_1 + 595x_2 - 524x_3 \equiv 516390040 \pmod{59515022(} \end{cases}$$

Вариант 5

$$\begin{cases} 20x_1 - 624x_2 - 1769x_3 \equiv 23570368 \pmod{23771221}; \\ 174x_1 - 836x_2 + 1782x_3 \equiv 51915582 \pmod{652163950}; \\ -1216x_1 + 421x_2 - 1334x_3 \equiv 213293056 \pmod{307945(} \end{cases}$$

Вариант 6

$$\begin{cases} 695x_1 + 1085x_2 - 3x_3 \equiv 551368951 \pmod{740177738}; \\ -847x_1 + 269x_2 + 307x_3 \equiv 533833234 \pmod{72658020} \\ -578x_1 + 1679x_2 + 419x_3 \equiv 68636756 \pmod{42349593} \end{cases}$$

Вариант 7

$$\begin{cases} -222x_1 + 457x_2 - 1834x_3 \equiv 327538372 \pmod{3282371(} \\ 236x_1 + 441x_2 - 1165x_3 \equiv 655627503 \pmod{903900358} \\ 656x_1 - 1178x_2 - 1602x_3 \equiv 775306899 \pmod{93013812} \end{cases}$$

Вариант 8

$$\begin{cases} -617x_1 + 1508x_2 + 64x_3 \equiv 64171388 \pmod{527835989)} \\ -101x_1 + 532x_2 + 774x_3 \equiv 158478963 \pmod{172590948} \\ 1160x_1 + 993x_2 + 824x_3 \equiv 207981078 \pmod{316465020} \end{cases}$$

Вариант 9

$$\begin{cases} -1961x_1 + 485x_2 - 84x_3 \equiv 607159408 \pmod{80602147(} \\ 16x_1 - 593x_2 - 1033x_3 \equiv 192347246 \pmod{368760563}; \\ -729x_1 + 1054x_2 - 1363x_3 \equiv 390552839 \pmod{688564(} \end{cases}$$

Вариант 10

$$\begin{cases} -1516x_1 + 36x_2 - 1761x_3 \equiv 212228239 \pmod{6493611(} \\ -1690x_1 - 1690x_2 + 1048x_3 \equiv 406012632 \pmod{714051} \\ 610x_1 - 691x_2 + 1015x_3 \equiv 657739743 \pmod{777338823} \end{cases}$$

Входные данные:система сравнений (согласно варианту).

Выходные данные: решение системы сравнений.

Содержание отчета: входные данные; выходные данные; программный код.

Контрольные вопросы:

1. Назовите команды для генерации сочетаний и перестановок, характерные для большинства современных математических пакетов?
2. Перечислите основные методы решения СЛАУ.

3. Приведите примеры современных математических пакетов?

4. Чем отличаются численные методы от символьных методов решения простейших математических задач?

5. Какие методы (символьные или численные) будут более эффективными при вычислении определенных и неопределенных интегралов? Ответ пояснить.

Лабораторная работа №2. Криптоанализ шифра простой замены.

Цель работы: выполнить криптоанализ текста, зашифрованного шифром простой замены с применением неперебoрного метода вскрытия шифротекста.

Теоретическая часть

Защита конфиденциальных сообщений в открытых сетях требует применения криптографических методов.

Криптография является самым мощным в настоящее время средством защиты информации и представляет самостоятельную прикладную науку, основанную на глубоких математических знаниях. Криптографические методы позволяют обеспечить конфиденциальность данных, устанавливать подлинность отправителя и контролировать целостность передаваемых сообщений.

Базовая модель криптографии, представленная на рис. 1 предполагает существование противника, имеющего доступ к открытому каналу связи и перехватывающего путем подслушивания все сообщения, передаваемые от отправителя к получателю. Подслушивание со стороны противника называется пассивным перехватом сообщений. Кроме того, противник может активно вмешивается в процесс передачи информации – модифицировать передаваемые сообщения, навязывать получателю свои собственные сообщения из канала. Такие действия называются активным перехватом сообщений.

Рис. 1. Базовая модель криптографии

Способы противодействия противнику основаны на применении методов шифрования. Все современные методы шифрования делятся на однолучевые или симметричные и двух ключевые или асимметричные.

В симметричных криптографических системах ключи, используемые на передающей и приемной сторонах, полностью идентичны. Такой ключ несет в себе всю информацию о засекреченном сообщении и поэтому не должен быть известен никому, кроме двух участвующих в разговоре сторон. Противник, наблюдая зашифрованное сообщение (шифротекст), не может прочитать сообщение. Отсюда возникает задача для противника – получение

открытого сообщения по наблюдаемому шифротексту, а именно задача сводится к раскрытию секретного ключа шифрования/дешифрования. Действия, предпринимаемые противником с целью раскрытия секретногоключа, называется атакой. Секретный канал, используемый для передачи секретного ключа от отправителя к получателю, должен исключать возможность утечки информации.

В асимметричных криптографических системах для шифрования и дешифрования применяются различные ключи. Для шифрования информации, предназначенной конкретному получателю, используют уникальный открытый ключ получателя-адресата. Соответственно для дешифрования получатель использует парный секретный ключ. Для передачи открытого ключа от получателя к отправителю секретный канал не нужен. Вместо секретного канала используется аутентичный канал, гарантирующий подлинность источника передаваемой информации (открытого ключа отправителя). Аутентичный канал является открытым и доступен противнику.

С помощью криптографических методов решается задача целостности информации. Целостность – это гарантия того, что информация сейчас существует в ее исходном виде, то есть при ее хранении или передаче не было произведено несанкционированных изменений. Целостность предполагает неизменность информационных объектов от их исходного состояния, определяемого автором или источником информации. Целостность является важнейшим аспектом информационной безопасности в тех случаях, когда информация используется для управления различными процессами, например, техническими, социальными и т.д. Так, ошибка в управляющей программе приведет к остановке управляемой системы, неправильная трактовка закона может привести к его нарушениям, точно так же неточный перевод инструкции по применению лекарственного препарата может нанести вред здоровью. Все эти примеры иллюстрируют нарушение целостности информации, что может привести к катастрофическим последствиям.

При обмене электронными документами по сети связи существенно снижаются затраты на обработку и хранение документов, убыстряется их поиск. Но при этом возникает проблема аутентификации автора документа и самого документа, то есть установления подлинности автора и отсутствия изменений в полученном документе. Эта задача также решается криптографическими методами, основанными на применении электронной цифровой подписи – ЭЦП. Электронная цифровая подпись используется для аутентификации текстов, передаваемых по телекоммуникационным каналам. ЭЦП решает проблему возможного спора между отправителем и получателем, в том числе и в суде, при наличии юридической базы для ее применения. В настоящее время криптографические методы нашли широкое применение не только для защиты информации от несанкционированного доступа, но и в качестве основы многих новых электронных

информационных технологий – электронного документооборота, электронных денег, тайного электронного голосования и др.

Самым простым из известных исторических шифров является шифр подстановки, который использовал Юлий Цезарь при переписке в военных походах.

В шифре Цезаря каждая буква алфавита заменяется буквой, которая находится на три позиции дальше в этом же алфавите. Рассмотрим на примере.

Открытый текст:meetmeafterthetogaparty

Шифрованный текст: PHHWPHDIWHUWKHWRJDSDUMB

Алфавит считается «циклическим» поэтому после Z идет A. Определить преобразование, можно перечислив все варианты, как показано ниже.

Открытый текст: abcdefghijklmnopqrstuvwxyz

Шифрованный текст: DEFGHIJKLMNOPQRSTUVWXYZABC

Если каждой букве назначить числовой эквивалент (

	dvvk	dv	rwkvi	kyv	kfxr	grikp
	dvvk	dv	rwkvi	kyv	kfxr	grikp
	cuuj	cu	gvjuh	jxu	jewq	fqhjo
	btti	bt	puitg	iwt	idvp	epgin
	assh	as	othsf	hvs	hcuo	dofhm
	zrrg	zr	nsgre	gur	gbtn	cnegl
	yggf	yg	mrfqd	ftq	fasm	bmdfk
	xppe	xp	lqepc	esp	ezrl	alcej
	wood	wo	kpdob	dro	dyqk	zkbdi
	vnnc	vn	jocna	cqn	cxpj	yjach
	ummb	um	inbmz	bpm	bwoi	xizbg
	tlla	tl	hmaly	aol	avnh	whyaf
	skkz	sk	glzkx	znk	zumg	vgxze
	rjjy	rj	fkyjm	ymj	ytlf	ufwyd
	qiix	qi	ejxiv	xli	xske	tevxc

Рис. 2. Криптоанализ шифра простой замены (шифра Цезаря) методом перебора всех вариантов ключей

Применение метода последовательного перебора всех возможных вариантов оправдано, если выполняются три важные характеристики данного шифра:

–известны алгоритмы шифрования и дешифрования;

–необходимо перебрать небольшое количество вариантов;

–язык открытого текста известен и легко узнаваем.

Алгоритм, для которого требуется перебрать слишком много ключей, делает криптоанализ на основе метода последовательного перебора практически бесполезным.

В этом случае для криптоаналитика существует другая линия атаки. Если криптоаналитик имеет представление о природе открытого текста (например, о том, что это несжатый текст на английском языке), можно использовать известную информацию о характерных признаках, присущих текстам на соответствующем языке. Методика криптоанализа текста, зашифрованного шифром простой замены с применением неперебoрного метода вскрытия шифротекста состоит из следующих этапов.

На первом этапе можно определить относительную частоту (вероятность)

Буква	$p(C)$	Буква	$p(C)$	Буква	$p(C)$
О	0,090	М	0,026	Й	0,010
Е	0,072	Д	0,025	Х	0,009
Ф	0,062	П	0,023	Ж	0,007
И	0,062	У	0,021	Ш	0,006
Н	0,053	Я	0,018	Ю	0,006
Т	0,053	З	0,016	Ц	0,004
С	0,045	Ы	0,016	Щ	0,003
Р	0,040	Б	0,014	Э	0,003
В	0,038	Ь,Ъ	0,014	Ф	0,002
Л	0,035	Г	0,013		
К	0,028	Ч	0,012		

Буква	$p(C)$	Буква	$p(C)$	Буква	$p(C)$
E	0,131	N	0,071	H	0,053
T	0,104	R	0,068	D	0,038
A	0,082	I	0,063	L	0,034
O	0,080	S	0,061	F	0,029
C	0,028	P	0,020	X	0,002
M	0,025	W	0,015	J	0,001
U	0,024	B	0,014	Q	0,001
G	0,020	V	0,009	Z	0,001
Y	0,020	K	0,004		

1.1. Выполнить частотный анализ символов зашифрованного текста.

1.2. Выписать символы из шифртекста с частотой большей 0,04 и произвести ранжирование их в порядке убывания частоты.

1.3. Сравнить ранги символов шифртекста с рангами символов в русском языке.

1.4. Выбрать 3–4 наиболее частых символа в шифртексте и русском языке с одинаковыми рангами.

1.5.Заменить три или четыре символа в шифртексте на символы русского языка с теми же рангами. Проверить совпадение рангов замененных символов в шифртексте и соответствующих символов в русском языке. Например, буква «о» самая частая буква в русском языке– она же должна оказаться самой частой буквой в шифртексте.

2.Использовать результаты грамматического анализа шифртекста и на его основе произвести полное дешифрование текста.

2.1. Выполнить предварительный анализ и расшифровку коротких слов (предлогов, местоимений, союзов междометий и т.п.) с использованием результатов, полученных в п. 1.5.

2.2. Для идентификации остальных символов шифртекста воспользоваться особенностями русского языка (удвоения букв, окончания слов и т.п.).

Содержание отчета

Отчет по лабораторной работе содержит следующие элементы.

- Результаты частотного анализа шифрованного текста для первых символов, имеющих частоту большую 0,04.
- Ранжирование наиболее часто встречающихся символов в порядке убывания.
- Таблица ранжирования символов, имеющих в расшифрованном тексте частоту большую 0,04, и сравнения с ранжированием наиболее частых символов русского языка.
- Примеры особенностей русского языка, использованных при дешифровании.
- Два-три варианта замены символов в шифртексте на символы русского языка с теми же рангами (попытки криптоанализа).
- Таблица замены наиболее часто встречающихся символов шифртекста на соответствующие им по рангу символы в русском языке.
- Ответ на контрольный вопрос, номер которого соответствует номеру варианта задания.
- Выводы по проделанной работе.

Контрольные вопросы

1.Что такое энтропия языка?

2.Что понимается под избыточностью сообщения?

3.Что такое шифр простой замены?

4.В чем состоит обобщение шифра Цезаря?

5.Опишите краткую историю возникновения шифра Цезаря.

6.Объясните принцип дешифрования шифра простой замены.

7.Объяснить, почему при вскрытии шифра простой замены используется не полное ранжирование по частоте всех символов русского языка, а лишь 3-4 наиболее частых символов, как в п. 1.4?

8.Какими характеристиками должен обладать шифр, чтобы была возможность применить метод частотного анализа?

9.Какие виды криптоанализа Вам известны?

10.Охарактеризуйте базовую модель криптографии.

11.Какие основные разновидности шифров простой замены применялись в прошлом?

12.Сформулируйте правила шифрования/дешифрования шифра Цезаря.

Лабораторная работа № 3. CloudSim: Платформа для моделирования и симуляции инфраструктур и сервисов облачных вычислений

Теоретическая часть

Недавно облачные вычисления стали ведущей технологией для предоставления надежных, безопасных, отказоустойчивых, устойчивых и масштабируемых вычислительных услуг, которые представлены как программное обеспечение, инфраструктура или платформа как услуги (SaaS, IaaS, PaaS). Более того, эти услуги могут предлагаться в частных центрах обработки данных (частные облака), могут предлагаться на коммерческой основе для клиентов (публичные облака) или, тем не менее, возможно, что как публичные, так и частные облака объединены в гибридные облака.

Эта и без того обширная экосистема облачных архитектур, наряду с растущим спросом на энергоэффективные ИТ-технологии, требует своевременных, повторяемых и контролируемых методологий для оценки алгоритмов, приложений и политик перед фактической разработкой облачных продуктов. Поскольку использование реальных испытательных стендов ограничивает эксперименты масштабом испытательного стенда и делает воспроизведение результатов чрезвычайно трудным делом, альтернативные подходы к тестированию и экспериментам используют развитие новых облачных технологий.

Подходящей альтернативой является использование инструментов моделирования, которые открывают возможность оценки гипотезы до разработки программного обеспечения в среде, в которой можно воспроизводить тесты. В частности, в случае облачных вычислений, где доступ к инфраструктуре предполагает платежи в реальной валюте, подходы на основе моделирования предлагают значительные преимущества, поскольку они позволяют клиентам облачных вычислений бесплатно тестировать свои услуги в повторяемой и контролируемой среде и настраивать производительность. узкие места перед развертыванием в реальных облаках. На стороне поставщика среды моделирования позволяют оценивать различные виды сценариев аренды ресурсов при различных распределениях нагрузки и цен. Такие исследования могут помочь поставщикам оптимизировать стоимость доступа к ресурсам с упором на повышение прибыли.

Основная цель работы - предоставить обобщенную и расширяемую структуру моделирования, которая обеспечивает бесшовное моделирование, симуляцию и экспериментирование с развивающейся инфраструктурой облачных вычислений и сервисами приложений. Используя CloudSim, исследователи и отраслевые разработчики могут сосредоточиться на конкретных проблемах проектирования системы, которые они хотят исследовать, не беспокоясь о деталях низкого уровня, связанных с облачными инфраструктурами и услугами.

Основные особенности
Обзор функций CloudSim:

- поддержка моделирования и симуляции крупномасштабных центров обработки данных облачных вычислений
- поддержка моделирования и симуляции виртуализированных хостов серверов с настраиваемыми политиками для предоставления ресурсов хоста виртуальным машинам
- поддержка моделирования и симуляции контейнеров приложений
- поддержка моделирования и симуляции энергоэффективных вычислительных ресурсов
- поддержка моделирования и симуляции сетевых топологий центров обработки данных и приложений для передачи сообщений
- поддержка моделирования и симуляции федеративных облаков
- поддержка динамической вставки элементов моделирования, остановки и возобновления моделирования
- поддержка определяемых пользователем политик выделения хостов виртуальным машинам и политик выделения ресурсов хоста виртуальным машинам

Задание.

1. Скачать пакет CloudSim, содержащий исходный код, примеры, jar-файлы и документацию по API, можно загрузить с веб-страницы CloudSim на GitHub: https://github.com/Cloudslab/cloudsim/releases
2. Создать центр обработки данных с одним хостом и запустить на нем один облачный пакет.
3. Создать два центра обработки данных с одним хостом и сетевой топологией каждый и запустить на них два облачных хранилища.
4. Создать два центра обработки данных с одним хостом в каждом и запустить облачные вычисления двух пользователей с сетевой топологией на них.
5. Создать два центра обработки данных с одним хостом в каждом и запустить на них два облачных хранилища.

Лабораторная работа№4. Оценка надежности распределенной системы хранения данных

Цель работы: построение и сравнение вероятностных моделей надежности распределенной системы хранения данных с использованием репликации и отказоустойчивого разделения данных.

Задание 1. Сгенерировать характеристики распределенной системы хранения данных (РСХД) и характеристики устройств хранения данных (жестких дисков – HardDiskDrive – HDD), на которых она построена (табл. 1).

Количество устройств хранения данных – целое число в диапазоне от 1 500 до 2 000 (генерируется случайно).

Время полного скраббинга данных хранилища – целое число в диапазоне от 2 до 14 (измеряется в сутках, генерируется случайно).

Коэффициент распределения Вейбулла (α) – величина в диапазоне от 0.7 до 0.8 (два разрядапосле точки, генерируется случайно).

Таблица 1 – Характеристики устройств хранения данных (жестких дисков)

#HDD	MTTF, млн. часов	Частота невосстановимых ошибок чтения, бит$^{-1}$	Скорость последовательного чтения с диска, Мб/с	Срок эксплуатации, суток	Средняя интенсивность доступа, %
1					
2					

MTTF (средняя наработка на отказ,MeanTimeToFailure) – величина в диапазоне от 1.2 до 1.5 (один разрядпосле точки, генерируется случайно).

Частота невосстановимыхошибокчтенияжесткого диска – величина равная 10^{-m}, где m– целое число из диапазона от 14 до 16 (генерируется случайно).

Скорость последовательного чтения с диска – целое число из диапазона от 120 до 190 (генерируется случайно).

Срок эксплуатации – величина, рассчитываемая на основе данных из журнала замены оборудования, от 0 до максимального срока службы жесткого диска, чаще всего – это 5 лет. В данной работе срок эксплуатации – целое число в интервале от 0 до 1825 (генерируется случайно). Отметим, что фактический максимальный срок службы жесткого диска обычно больше 5 лет, это связано с интенсивностью доступа к данным на жестком диске: средняя загруженность жесткого диска колеблется в интервале от 20% до 90%, тогда как срок службы, равный 5 годам, рассчитан для предполагаемой средней загруженности 100%.

Средняя интенсивность доступа – целое число из диапазона от 20 до 90 (измеряется в %, генерируется случайно).

Входные данные:–.

Выходные данные: количество устройств хранения данных; время полного скраббинга данных хранилища; коэффициент распределения Вейбулла; таблица 1 – характеристик устройств хранения данных.

Содержание отчета: выходные данные; программный код.

Задание 2. Рассчитать показатели AFRи iredля каждого из устройств хранения на основе данных таблицы 1. Сформировать таблицу 2, содержащую значения рассчитанных вероятностных показателей для каждого из устройств хранения. Размер одного блока данных – 64 Мб.

Таблица 2 – Вероятностные показатели для каждого из устройств хранения

#HDD	ARF	ire
1		
2		

Входные данные:размер одного блока данных; время полного скраббинга данных хранилища; коэффициент распределения Вейбулла; таблица 1 – характеристик устройств хранения данных.

Выходные данные: таблица 2 – значений рассчитанных вероятностных показателей для каждого из устройств хранения.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 3. Смоделировать процесс хранения файловразличного размера: менее 64 Мб, 64-128 Мб и т.д. до 20 Гб с шагом 64 Мб. Построить модели надежности распределенной системы хранения данных, использующие репликацию и отказоустойчивое разделение данных. Выполнить следующие этапы при построении моделей:

1) Рассчитать количество блоков после разрезания файла.

2) Распределить блоки между необходимым количеством жестких дисков (номера дисков #HDDгенерируются случайно).

3) Рассчитать вероятность отказа системы при запросе (ProbabilityofFailureonDemand – PFD), учитывая показатели AFRи irекаждого из запоминающих устройств, используемых для хранения файла.

При расчете показателя PFD, повторить эксперимент не менее 20 раз, сохранить в таблице 3 минимальное, среднее и максимальное значения PFD (за 20 циклов), значение фактора репликации (*RF*) и параметры схемы (*k*, *n*) отказоустойчивого разделения данных для файловразличного размера. При выборе схемы (*k*, *n*) отказоустойчивого разделения данных использовать в качестве порога, т.е. количества блоков, достаточного для восстановления файла, значение 4 ($k = 4$), т.е. начальные значения: $RF = 1$, $(k, n) – (4, 4)$.

Фактор репликации и схему отказоустойчивого разделения данных выбрать таким образом, чтобы средняя вероятность отказа системы при запросе (PFD)не превышала 10^{-2}.

Таблица 3 – Характеристики моделей надежности хранения для файлов различного размера

Размер файла	*CC*	PFD_{min}	PFD_{max}	PFD	*RF*	(*k*, *n*)
менее 64 Мб						
64-128 Мб						
…						
20 Гб						

Входные данные:таблица 2 – значений рассчитанных вероятностных показателей для каждого из устройств хранения.

Выходные данные: таблица 3 – характеристик моделей надежности хранения для файлов различного размера.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 4. Рассчитать и сохранить в таблице 4 значения избыточности данных для файлов различного размера при использовании репликации и отказоустойчивого разделения данных (строки с одинаковыми парами значений $Redundancy_{RF}$ и $Redundancy_{(k, n)}$ сохранять не нужно). Отобразить полученные результаты на графике: полуось *Ox*– количество блоков (*CC*), полуось *Oy*– избыточность (Redundancy). Сделать вывод.

Таблица 4 – Характеристики моделей надежности хранения

Размер файла	*CC*	*RF*	(*k*, *n*)	$Redundancy_{RF}$	$Redundancy_{(k, n)}$
менее 64 Мб					
64-128 Мб					
…					
20 Гб					

Входные данные:таблица 3 – характеристик моделей надежности хранения для файлов различного размера.

Выходные данные: таблица 4 – характеристик моделей надежности хранения.

Содержание отчета: входные данные; выходные данные; график; программный код.

Задание 5*. Установить максимально достижимый уровень надежности для двух моделей распределенного хранения (основанной на репликации и основанной на отказоустойчивом разделении данных), т.е. установить наибольшее значение *tr_max*(maximumthreshold), еслиPFD = 10^{-tr_max}.

* – задание не является обязательным для выполнения.

Контрольные вопросы:

1. Что отражает и как рассчитывается показатель ire?
2. Почему реальный срок службы жестких дисков, используемых в современных РСХД, зачастую выше заявленного в техническом описании данных устройств?
3. Что отражает коэффициент распределения Вейбулла? Какие значения коэффициента распределения Вейбулла характерны для современных РСХД?
4. Что нужно сделать для повышения надежности хранения данных при использовании модели с репликацией?
5. Что нужно сделать для повышения надежности хранения данных при использовании модели с отказоустойчивым разделением данных?

Лабораторная работа №5. Модулярная арифметика

Цель работы: изучение программных средств реализации основных операций модулярной арифметики, разработка программных модулей кодирования/декодирования в/из систему остаточных классов.

Формула для выбора варианта:

Вариант = ((номер по списку) mod (количество вариантов)) + 1.

Задание 1. Разработать программный модуль для выбора компактной последовательностиоснований системы остаточных классов (СОК) согласно варианту (табл. 1).

Входные данные: b – разрядность чисел, представляемых в СОК; n – количество оснований СОК.

Выходные данные: $\{m_i\}$ – набор оснований СОК, $i = 1..n$.

Содержание отчета: входные данные; условное среднее значение модуля СОК; набор оснований, не содержащий/содержащий степень двойки; разрядность наибольшего из оснований в наборе не содержащем/содержащем степень двойки; выходные данные (выбранный набор оснований СОК); программный код.

Задание 2. Разработать программный модуль для кодирования данных в выбранной СОК. Использовать значение параметра разбиения (окна) sсогласно варианту (табл. 1). Сгенерировать случайное b-битное целое число X (bвыбирается согласно варианту из таблицы 1). Получить представление X в выбранной СОК.

Входные данные: s – параметр разбиения; b– разрядность целого число, переводимого в СОК;X – b-битное целое число, представленное в позиционной десятичной системе счисления.

Выходные данные: $X = (x_1, x_2, \ldots, x_n)$ – b-битное целое число, представленное в выбранной СОК.

Содержание отчета: входные данные; LUT-таблицы по каждому из оснований; выходные данные; программный код.

Задание 3. Разработать программный модуль для декодирования данных из СОК, используя один из методов:

– основанный на Китайской теореме об остатках (КТО);

– основанный на переходе к представлению данных в обобщенной позиционной системе счисления (ОПСС);

– основанный на Китайской теореме об остатках с дробными величинами (КТОд);

согласно варианту (табл. 1). Получить позиционное представление X.

Входные данные: $X = (x_1, x_2, \ldots, x_n)$ – b-битное целое число, представленное в выбранной СОК.

Выходные данные: X – b-битное целое число, представленное в позиционной десятичной системе счисления.

Содержание отчета: входные данные; константы метода декодирования; LUT-таблицы для умножения на константы по каждому из оснований; выходные данные; программный код.

Таблица 1 – Варианты заданий

Вариант	b, бит	n	s, бит	Метод декодирования
1	32	4	2	КТО
2	32	4	2	ОПСС
3	32	4	2	КТОд
4	32	4	4	КТО
5	32	4	4	ОПСС
6	32	4	4	КТОд
7	32	4	8	КТО
8	32	4	8	ОПСС
9	32	4	8	КТОд
10	32	6	2	КТО
11	32	6	2	ОПСС
12	32	6	2	КТОд
13	32	6	4	КТО
14	32	6	4	ОПСС
15	32	6	4	КТОд
16	32	6	8	КТО
17	32	6	8	ОПСС
18	32	6	8	КТОд
19	64	4	2	КТО
20	64	4	2	ОПСС
21	64	4	2	КТОд
22	64	4	4	КТО
23	64	4	4	ОПСС
24	64	4	4	КТОд
25	64	4	8	КТО
26	64	4	8	ОПСС
27	64	4	8	КТОд
28	64	6	2	КТО
29	64	6	2	ОПСС
30	64	6	2	КТОд
31	64	6	4	КТО
32	64	6	4	ОПСС
33	64	6	4	КТОд
34	64	6	8	КТО
35	64	6	8	ОПСС
36	64	6	8	КТОд

Контрольные вопросы:

1. Почему при кодировании данных в СОК не используется деление с остатком?
2. На что влияет значение параметра разбиения (окна) при кодировании данных в СОК?
3. Охарактеризуйте и сравните два метода перехода от представления данных в СОК к представлению в ОПСС: метод, основанный на схеме Гарнера, и метод, основанный на использовании ортогональных базисов КТО, представленных в ОПСС.
4. Какое условие должно соблюдаться для корректного перехода от представления данных в СОК к представлению в ОПСС?
5. В чем принципиальное отличие декодирования данных из СОК на основе КТО от декодирования с помощью КТОд?

Лабораторная работа №6. Методы и алгоритмы исправления ошибок данных

Цель работы: разработка программных модулей отказоустойчивого кодирования данных и декодирования ошибок данных разного типа.

Формула для выбора варианта:

Вариант = ((номер по списку) mod (количество вариантов)) + 1.

Задание 1. Разработать программный модуль для восстановления данных в случае t-кратного аппаратно-программного сбоя, основанного на использовании (n, k)избыточной системы остаточных классов (ИСОК).

Входные данные: k – количество рабочих модулей ИСОК (или пороговое значение для количества подблоков, минимально необходимого для восстановления блока данных); t – количество декодируемых аппаратно-программных сбоев; $\{e_1, e_2, \ldots, e_t\}$ – номера позиций подблоков, недоступных в результате аппаратно-программных сбоев; тестовые цифровые данные – число X, в десятичной системе счисления.

Выходные данные: n – количество модулей (оснований) ИСОК; $\{m_i\}$ – набор оснований ИСОК, $i = 1..n$; тестовое восстановленное число X, в десятичной системе счисления.

Содержание отчета: входные данные; метод, используемый при восстановлении; выходные данные; программный код.

Задание 2. Разработать программный модуль для исправления однобитовых ошибок данных, основанный на использовании (n, k)циклическогокода Хэмминга.

Входные данные: b – разрядность кодируемого сообщения; тестовые цифровые данные – b-разрядное число X, в десятичной системе счисления.

Выходные данные: (n, k)-схемациклического кода Хэмминга; тестовое восстановленное число X, в десятичной системе счисления.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 3. Разработать программный модуль для восстановления данных в случае t-кратной пакетной ошибки, основанного на методе проекций с максимальным правдоподобием для (n, k)-ИСОК.

Входные данные: k – количество рабочих модулей ИСОК (или пороговое значение для количества подблоков, минимально необходимого для восстановления блока данных); t – количество декодируемых пакетных ошибок; тестовые цифровые данные – число X, в десятичной системе счисления.

Выходные данные: n – количество модулей (оснований) ИСОК; $\{m_i\}$ – набор оснований ИСОК, $i = 1..n$; тестовое восстановленное число X, в десятичной системе счисления.

Содержание отчета: входные данные; метод, используемый при восстановлении; выходные данные; программный код.

Контрольные вопросы:

1. Перечислите основные этапы процедуры исправления ошибок данных?
2. Чем процедура декодирования пакетной ошибки данных отличается от процедуры декодирования ошибки данных, связанной с аппаратно-программным сбоем?
3. Предложите любую (n, k)-схему отказоустойчивого разделения данных,способную декодировать ошибку, возникшую в результате 4-ех аппаратно-программных отказов и 2-ух пакетных ошибок данных.
4. В чем заключается основная идея классического метода проекций в случае ИСОК?
5. Какая основанная идея лежит в основе концепции декодирования с максимальным правдоподобием?

Лабораторная работа №7. Методы и алгоритмы сжатия данных без потерь

Цель работы: реализация алгоритмов, характерных для основных классов методов сжатия данных без потерь.

Задание 1. Разработать программный модуль для сжатия данных алгоритмом RLEи алгоритмом RLEс преобразованием BWT.

Входные данные: тестовые символьные данные.

Выходные данные: сжатые тестовые данные.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 2. Разработать программный модуль для сжатия данных алгоритмом арифметического кодирования.

Входные данные: тестовые символьные данные.

Выходные данные: сжатые тестовые данные.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 3. Разработать программный модуль для сжатия данных алгоритмом LZ78.

Входные данные: тестовые символьные данные.

Выходные данные: сжатые тестовые данные.

Содержание отчета: входные данные; выходные данные; программный код.

Задание 4. Оценить качество сжатия для всех реализованных алгоритмов, построить сравнительные графики для коротких (до 500 000 символов), средней длины (2 000 000-5 000 000 символов) и длинных (более 50 000 000 символов) символьных последовательностей: полуось *Ox*– количество символов, полуось *Oy*– качество сжатия. В качестве тестовых символьных последовательностей использовать отрывки из любого художественного произведения.

Входные данные: тестовые символьные данные.

Выходные данные: сжатые тестовые данные; коэффициент сжатия.

Содержание отчета: входные данные; графические данные; выходные данные; программный код.

Контрольные вопросы:

1. Какая идея лежит в основе любого алгоритма сжатия данных?
2. От чего зависит качество сжатия данных?
3. На какие два класса можно разделить алгоритмы сжатия данныхбез потерь?

4. Что означает аббревиатура RLE?
5. Расшифруйте аббревиатуру BWT.

Лабораторная работа № 8 Манипуляция данными в системах распределенных баз данных

Цель работы. Изучение методов связывания объектов и манипуляций данными базы данных, распределенной на нескольких компьютерах сети.

Теоретическая часть

Большинство систем РД состоит из нескольких баз данных, управляемых разными серверами, расположенными в различных местах. Все серверы и клиенты Oracle должны использовать Net8 - сетевое программное средство Oracle для взаимодействия друг с другом по сети. Каждый сервер базы данных в РБД управляет доступом к своей локальной базе данных – за управление системой в целом не отвечает ни один сервер.

Именование объектов

Все сервисы, доступные в сети, должны иметь уникальные имена, чтобы пользователи и приложения знали, как с ним обращаться. Глобальное имя базы данных состоит из двух частей:

- основное имя базы данных, назначаемое ей при создании. Имя базы данных не должно содержать больше восьми символов.
- сетевой домен базы данных, который показывает логическое местонахождение базы данных в сети.

На рисунке 2 представлена сеть баз данных гипотетической компании SALESMENT (продажи). Сеть SALESMENT состоит из трех баз данных WS1, WS2 и WS3. Им соответствуют глобальные имена (имена сервисов) WS1. SALESMENT, WS2. SALESMENT, WS3. SALESMENT.

Для того, чтобы ссылаться на конкретные имена объектов схемы базы данных, не являющейся локальной, нужно дополнить имя объекта глобальным именем базы данных. На рисунке 2 показано, что в каждой из баз данных WS1, WS2 и WS3 содержится таблица PROVIDER (производитель/поставщик).

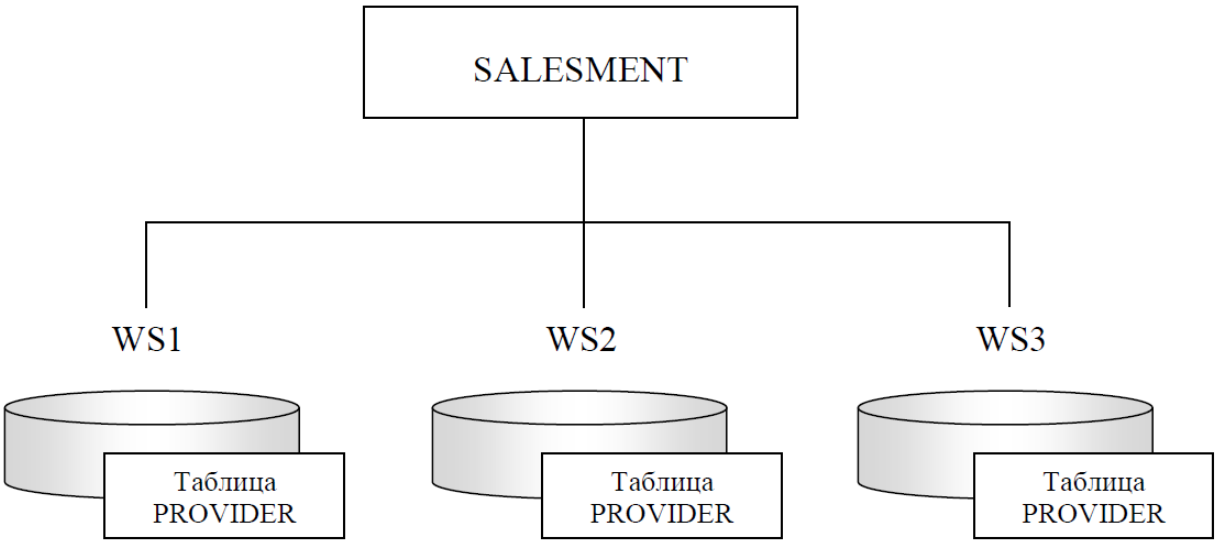

Рисунок 2 – Структура сети баз данных SALESMENT

Если запустить приложение (например, SQL*Plus) и соединиться с базой данных (например, WS2), то можно обратиться как к таблице PROVIDER базы данных WS2, так и к таблице PROVIDER базы данных WS1, идентифицировав этот объект с помощью его составного имени в распределенной базе данных:

*SELECT * FROM provider@ws1.salesment;*

Выполняя этот запрос, сервер локальной базы WS2 неявно использует связь баз данных, соединяющую базы данных WS1 и WS2.

Прозрачность распределенной базы данных

Пользоваться указанной системой именования возможно при разработке очень небольших баз данных, т.к. каждый разработчик должен знать текущее местоположение объектов в системе распределенной базы данных.

В Oracle имеется средство, позволяющее сделать работу с этими объектами прозрачной. Это механизм создания синонима, который скрывает физическое место хранения объекта в системе распределенной базы данных:

CREATE PUBLIC SYNONIM prov1
FOR provider@ws1.salesment;

После создания общего синонима пользователи локальной базы данных могут ссылаться на удаленную таблицу PROVIDER рабочей станции WS1 как на локальную. Oracle автоматически превращает локальный псевдоним в имя удаленной таблицы и использует для обращения к ней связь базы данных.

*SELECT * FROM prov1;*

Другим средством прозрачности могут служить представления. Например, локальное представление PRODUCT указывает на данные, содержащиеся в удаленной таблице PRODUCT рабочей станции WS1.

CREATE VIEW product AS
*SELECT * FROM product@ws1.salesment;*

Установление связей баз данных

Oracle предлагает такую поддержку распределенных баз данных, которая позволяет производить удаленные и распределенные запросы по каналам связи баз данных.

Для того чтобы предоставить доступ к удаленным базам данных в распределенной системе, необходимо установить в локальной баз данных связи баз данных (databaselink). Связь двух баз данных обозначает однонаправленную линию связи между двумя базами данных Oracle (рисунок 3).

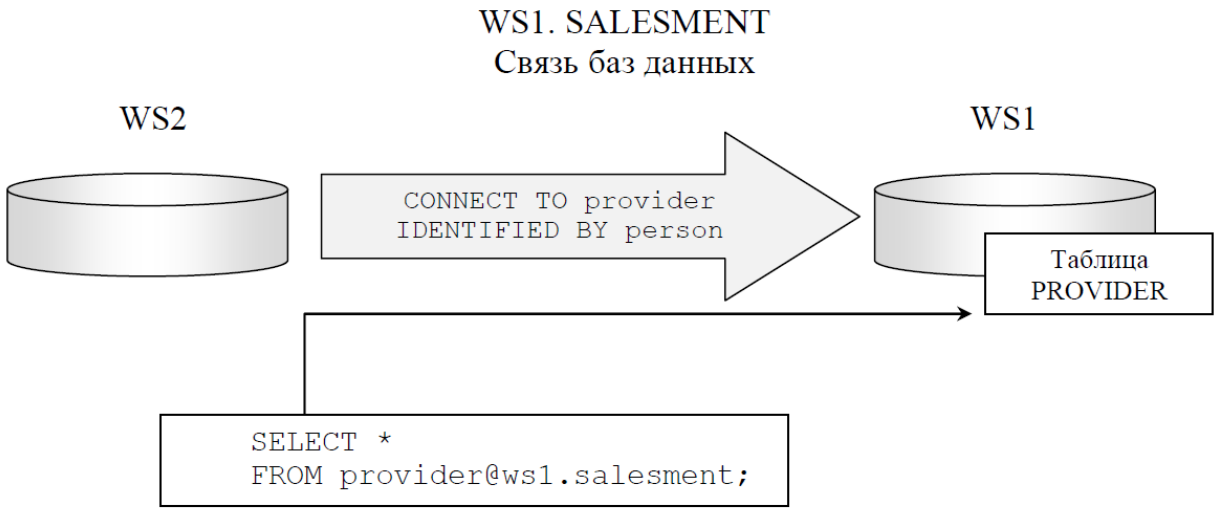

Рисунок 3 – Пример связи баз данных Oracle

Установление связей можно использовать для достижения двоякой цели — обеспечения прозрачности расположения и независимости расположения. Например, с помощью следующего оператора в локальной базе WS2 можно создать связь, описывающую путь к удаленной базе данных WS1. SALESMENT:

CREATE DATABASE LINK ws1.salesment;

При разработке приложений для работы в распределенной базе данных необходимо выполнять различные операции манипуляций с данными: удаленные запросы, обновления и т.д. Рассмотрим примеры операторов SQL и PL/SQL, позволяющие выполнить эти операции.

Установив связи между базами данных, программа может манипулировать данными на любом из узлов сети.

Удаленные запросы

Удаленный запрос (remotequery) – это оператор SELECT, считывающий информацию в нескольких удаленных таблицах, находящихся в одном и том же удаленном узле. Например, с помощью следующего удаленного запроса информация считывается в удаленных таблицах PROVIDER и PRODUCT базы данных WS3.

SELECT prov.name, p.name, p.price
FROM provider@ws3.salesment prov, product@ws3.salesment p;

Распределенные запросы

Распределенный запрос (distributedquery) – это оператор SELECT, считывающий информацию из двух или более баз данных. Например, с помощью следующего распределенного запроса информация считывается в локальной таблице PROVIDER и удаленной таблице PRODUCT базы данных WS3.

SELECT prov.name, p.name, p.price
FROM provider prov, product@ws3.salesment p;

Удаленное обновление

Удаленное обновление (remoteupdate) – это операция обновления, с помощью которой модифицируются данные удаленной таблицы. Например, с помощью следующего оператора UPDATE обновляется строка в таблице PRODUCT удаленной базы данных WS1.

UPDATE product@ws1.salesment
SET prod_price = 200
WHEREprod_id = 1;

Распределенное обновление

Распределенное обновление (distributedupdate) модифицирует данных двух и более серверов. Единственный способ распределенного обновления – создать хранимую процедуру, метод объектных типов и т.д. и включить их в состав две или более операции обновления, каждая из которых обновляет данные в различных базах данных. Например, следующий анонимный блок PL/SQL можно считать распределенным обновлением:

BEGIN
UPDATE product
SET prod_price = 200
WHERE prod_id = 1;
UPDATE product@ws2.salesment
SET prod_price = 200
WHERE prod_id = 1;
UPDATE product@ws3.salesment
SET prod_price = 200
WHEREprod_id = 1;
END;

Удаленные транзакции

Удаленная транзакция (remotetransaction) – это транзакция, содержащая один или более удаленных операторов, каждый из которых ссылается на одну и ту же удаленную базу данных. Например, следующая удаленная транзакция обновляет информацию только в базе данных WS1.

UPDATE product@ws1.salesment
SET prod_price = 200
WHERE prod_id = 1;
UPDATE product@ws1.salesment
SET prod_price = 300
WHERE prod_id = 2;
UPDATE product@ws1.salesment
SET prod_price = 150
WHEREprod_id = 3;
COMMIT;

Распределенная транзакция (distributedtransaction) – это транзакция, включающая один или более операторов, обновляющих информацию в двух и более разных базах данных. Например, следующая распределенная транзакция обновляет информацию в нескольких базах данных.

```
UPDATE product
        SET prod_price = 200
        WHERE prod_id = 1;
UPDATE product@ws1.salesment
        SET prod_price = 200
        WHERE prod_id = 1;
UPDATE product@ws3.salesment
        SET prod_price = 150
        WHERE prod_id = 3;
COMMIT;
```

Двухфазнаяфиксация

При установлении соединений программа может обновлять данные где угодно. Однако она также должна выдать явную команду фиксации в каждый из экземпляров, в который она выдала DML-команды, иначе в какой-то момент времени разные пользователи могут видеть рассогласованное состояние базы данных.

Для транзакции, как единого целого, должна быть выполнена либо полная фиксация (операция завершения), либо полный откат. Чтобы обеспечить соблюдение этого правила для распределенных транзакций, в Oracle применяется специальный алгоритм двухфазного завершения (two-phasecommitmechanism), который координирует управление транзакциями в сети. Двухфазное завершение необходимо при сетевых и системных сбоях, которые могут прерывать завершение распределенных транзакций.

В общих чертах действие этого механизма выглядит следующим образом: когда выдается команда фиксации, один из экземпляров базы данных, участвующий в транзакции, принимает на себя роль координатора. На этапе первой фазы этот экземпляр выбирает один из экземпляров в качестве точки фиксации и дает всем остальным задействованным экземплярам (среди которых может быть и он сам, если он не является точкой фиксации) указание приготовиться. После того как получено подтверждение на фиксацию от других экземпляров, точке фиксации предлагается выполнить обычную фиксацию. В зависимости от результата этой фиксации координатор просит все остальные экземпляры либо зафиксировать транзакцию, либо выполнить ее откат.

Конечно, и в этом случае возможны всякого рода сбои. В любой момент могут отказать как отдельные серверы, так и сетевые каналы связи, но, невзирая ни на что, программное обеспечение должно гарантировать целостность данных. В итоге не только возникает значительный трафик сообщений, но и создаются потенциально устойчивые блокировки на уровне блоков.

Механизмы двухфазного завершения являются внутренними процессами серверами баз данных Oracle, участвующих в транзакции. Пользователь должен лишь закончить распределенную транзакцию оператором COMMIT; остальную работу выполняет Oracle.

Закрытие каналов связи

Для "популярных" серверов, т.е. серверов, которые являются объектом множества открытых каналов связи БД, число одновременно открытых Oracle-соединений может стать таким большим, что вызовет чрезмерную подкачку страниц памяти. Последние версии Oracle не только позволяют администратору устанавливать, какое максимальное число каналов связи БД может быть открыто для процесса (через параметр DB_LINKS в файле init.ora), но и разрешают сеансу закрывать канал связи БД, занимающий ресурсы на удаленном сервере. Воткомандыдляэтогопримера:

ALTER SESSION CLOSE DATABASE LINK ws1.salesment;

ALTER SESSION CLOSE DATABASE LINK ws2.salesment;

ALTER SESSION CLOSE DATABASE LINK ws2.salesment;

К сожалению, для установления соединения с Oracle необходимы большие затраты времени центрального процессора на серверной стороне канала, поэтому при сколько-нибудь значительной вероятности того, что этот канал в ближайшем будущем понадобится вновь, вряд ли эффективно его закрывать. "Замораживание" одного-двух мегабайтов памяти на удаленном сервере может оказаться меньшим злом, чем затраты на подключение и отключение, которые придется понести потом. Если же удаленные серверы имеют ограниченные ресурсы и известно, что какие-либо каналы вряд ли еще будут использоваться, то их закрытие позволит увеличить объем доступных ресурсов. Однако при этом также разрушится прозрачность расположения и потребуется более богатое функциональными возможностями приложение, чем то, которое захотят реализовать большинство проектировщиков и программистов.

Варианты увеличения производительности при распределенных соединениях

Распределенные соединения могут создавать проблемы с производительностью все время, пока выполняются операции манипулирования данными в распределенных базах данных.

Если производительность неудовлетворительна, можно рассмотреть следующие варианты:

• На ранних стадиях проекта необходимо постоянное тестирование распределенного соединения на конкретных примерах.

• Создать соединение так, чтобы обеспечить ссылку на одно или несколько представлений и переместить ключевые элементы оптимизации запросов на серверы, где их можно оптимизировать с большей эффективностью.

• Выполнить соединение средствами приложения с использованием удаленного SQL. Эта стратегия может быть эффективной, если удаленные данные можно получить за один запрос или за очень малое число таких запросов.

• Изменить распределение данных, использованных в соединении, перенеся одну или несколько таблиц в другую базу данных (или каким-либо образом реплицировав эти данные). Это решение требует фундаментального

изменения в структуре, и для поиска эффективной стратегии распределения, возможно, потребуется перебрать несколько вариантов.

Последнее решение подводит нас к сути проектирования распределенных баз данных:

Стратегия распределения данных должна определяться требованиями к производительности и работоспособности приложения, т.е. данные должны быть распределены по сети таким образом, чтобы максимально соответствовать запросам информационной системы, использующей эти данные.

Задание

1. Установить связи между локальными базами данных WS1, WS2, WS3.
2. Пользователю, выбранному в качестве администратора, создать синонимы для локальных базам данных WS1, WS2, WS3.
3. Создать представления на основе локальных и удаленных таблиц.
4. Проверить работоспособность созданных представлений командой SQL Select.
5. Выполнить удаленный запрос, удаленное обновление, удаленное добавление, удаленное удаление. При выполнении используйте детальный контроль доступа (WHERE).
6. Выполнить распределенный запрос, распределенное обновление, распределенное добавление, распределенное удаление. При выполнении используйте детальный контроль доступа (WHERE).
7. Выполнить удаленную транзакцию.
8. Выполнить распределенную транзакцию.
9. Оформить отчет о выполнении лабораторной работы.

Содержание отчета

1. Цель работы.
2. Операторы SQL, позволяющие создавать синонимы, представления, связи в системе распределенной базы данных.
3. Операторы SQL, позволяющие выполнить удаленные и распределенные манипуляции с данными.
4. Структура физических таблиц Oracle.

Контрольные вопросы по лабораторной работе:

1. Каким образом именуются объекты в распределенной базе данных?
2. Средства Oracle для обеспечения прозрачности имен объектов в распределенной базе данных.
3. Отличие локальной, удаленной и распределенной обработки.
4. Механизм двухфазной фиксации.

Лабораторная работа № 9. Гомоморфное шифрование

Цель работы: Реализация гомоморфных схем шифрования

Теоретическая часть

Конечная цель с различными схемами HE - получить неограниченную и практичную схему FHE. Схемы PHE и схемы SWHE, предложенные до революционной работы Gentry в 2009 году, были ступеньками к этой цели. Тем не менее, они ограничены с точки зрения областей применения. Однако схемы SWHE, предложенные после работы Джентри, в основном являются частью схем FHE, а не другой схемой. Более того, ограниченный (уровневый) FHE также можно назвать схемой SWHE. Следовательно, невозможно разделить схемы SWHE и FHE для работ, предложенных после работы Джентри. В этом разделе мы суммируем реализации схем SWHE и FHE, которые могут привести к новым работам и ускорить последующие работы, предложенные после работы Джентри.

Внедрение криптографической схемы - это средний шаг между ее разработкой и ее применением в реальной службе, и он обеспечивает реалистичную оценку производительности разработанной схемы. Хотя некоторые новые предложенные схемы FHE значительно повысили эффективность и производительность реализаций, накладные расходы и стоимость реализаций FHE все еще слишком высоки, чтобы их можно было прозрачно применять в реальных услугах, не беспокоя пользователя.

«Полностью» реализованные схемы FHE

После решения долгосрочной открытой проблемы разработки полностью гомоморфной схемы (Gentry 2009) многие новые предложения полностью гомоморфных схем были протестированы с реализациями. В самой первой попытке Смарт и Веркаутерен (2010) реализовали вариант первоначальной схемы Джентри. Однако их генерация ключей занимает часы до

сгенерирована за 2,2 часа, а размер открытого ключа составил 2,25 ГБ. На расшифровку зашифрованных текстов (самозагрузку) ушла 31 минута. После этого в Coronetal. (2011), целочисленный вариант схемы FHE, первоначально представленный Van Dijk et al. (2010 г.). В этой реализации генерация ключа занимает 43 минуты, а размер открытого ключа составляет 802 МБ. Реализация показала, что того же уровня безопасности можно достичь с помощью гораздо более простой схемы. (Разница заключается в разных определениях уровней безопасности.) Позже Coronetal. в другой работе (Coronetal. 2012) увеличен размер открытого ключа до 10 МБ, генерация ключа до 10 минут и процедура шифрования до 11 минут и 34 секунд с использованием аналогичных настроек параметров в Coronetal. (2011). Эта производительность достигается с помощью метода сжатия открытого ключа. В Coronetal. (2012), выровненная схема DGHV также реализована с несколько худшими характеристиками. ЮанмиЧен и Фонг К. Нгуен (ChenandNguyen 2012) предложили алгоритм для разрушения схемы в Coronetal. (2012), что быстрее, чем полный поиск. Эта работа показала, что уровень безопасности схемы, предложенной Coronetal. (2012) намного ниже, чем схема, предложенная в GentryandHalevi (2011).

Таблица 3. «Полностью» реализованные схемы FHE

Scheme Information		Platform	Parameters		Running Times				
Implemented Scheme	Base Scheme	Software	Security Parameter, λ	Dimension, n	PK Size	KeyGen	Enc	Dec	Recrypt
GH11 (Gentry and Halevi 2011)	Gen09 (Gentry 2009)	C/C++	72	33,768	2.25GB	2.2 h	3 min (SWHE)	0.66 s (SWHE)	31 min
CMNT11 (Coron et al. 2011)	DGHV10 (Van Dijk et al. 2010)	Sage 4.5.3	72	7,897	802MB	43 min	2 min 57 s	0.05 s	14 min 33 s
CNT12 (with compressed PK) (Coron et al. 2012)	DGHV10 (Van Dijk et al. 2010)	Sage 4.7.2	72	7,897	10.3MB	10 min	7 min 15 s	0.05 s	11 min 34 s
CNT12 (leveled) (Coron et al. 2012)	DGHV10 (Van Dijk et al. 2010)	Sage 4.7.2	72	5,700	18MB	6 min 18 s	3.4 s	0.00 s	2 h 27 min

Реализация FHE для схем «малой глубины» (“Low-Depth”).

Второй тип реализаций FHE пытался реализовать выровненные схемы FHE для схем малой глубины с заданным временем выполнения для изолированного и составного сложения и умножения (Naehriget al.2011; Boset al.2013; LepointandNaehrig 2014; RohloffandCousins 2014).). Сравнение этих реализаций FHE с малой глубиной дано в таблице 4. Поскольку производительность современного уровня техники была неудовлетворительной, в качестве первой попытки в Naehrig был реализован относительно более простой FHE, который допускает только несколько операций гомоморфного умножения. и другие. (2011). Позже эта производительность была улучшена Bosetal. (2013) из-за нового метода оценки операции гомоморфного умножения. Более того, в отличие от (Naehrigetal. 2011), в Bosetal. (2013) основная схема была реализована на языке программирования C, чтобы избежать нежелательных накладных

расходов из-за системы компьютерной алгебры. Затем аналогичная работа с Bosetal. (2013). Недавно было сделано значительное улучшение за счет использования двойной CRT в представлении зашифрованных текстов и использования параллелизма для ускорения реализации в Matlab (RohloffandCousins 2014).

Таблица 4. Реализации FHE для схем «малой глубины»

Scheme Information		Platform	Parameters		Running Times			
Implemented Scheme	**Base Scheme**	**Software**			**Enc**	**Dec**	**Mult**	**Add**
NLV11 (Naehrig et al. 2011)	BV11 (Brakerski and Vaikuntanathan 2011)	Magma	$w = 2^{32}$	q=127	756ms	57ms	1,590ms	4ms
YASHE (by BLLN13 (Bos et al. 2013))	LTV12 (López-Alt et al. 2012)	C/C++	$t = 2^{10}$	q=130	27ms	5ms	31ms	0.024ms
YASHE (by LN14a (Lepoint and Naehrig 2014))	LTV12 (López-Alt et al. 2012)	C/C++	$w = 2^{32}$	q=130	16ms	15ms	18ms	0.7ms
FV (by LN14a (Lepoint and Naehrig 2014))	BV11 (Brakerski and Vaikuntanathan 2011)	C/C++	$w = 2^{32}$	q=130	34ms	16ms	59ms	1.4ms
RC14 (Rohloff and Cousins 2014)	LTV12 (López-Alt et al. 2012)	Matlab	$n = 2^{10}$	t=1	12ms	3.36ms	100ms	0.56ms

«Реальные» сложные реализации FHE.

В отличие от приведенных выше схем, которые являются либо проверкой концепции, либо реализациями малой глубины, авторы Gentry et al. (2012) впервые реализовали

2014), SIMON (Lepoint и Naehrig, 2014) и LowMC (Альбрехт и др., 2015). Мелла и Суселла (2013) оценили стоимость некоторых симметричных криптографических примитивов, таких как хэш-функция AES-128, SHA-256, потоковый шифр Salsa20 и губчатая функция KECCAK. Они пришли к выводу, что AES лучше всего подходит для гомоморфной оценки из-за небольшого количества циклов и отсутствия целочисленных операций и логических операций AND во внутреннем устройстве. Однако в MellaandSusella (2013) реализован только AES-128.

Таблица 5. Комплексные реализации FHE в «реальном мире»

Scheme			Platform	Parameters		Running Times		
Implemented Scheme	Base Scheme	Circuit	Reported Specs	λ	AND Depth	Total Evaluation Time	Number of Parallel Enc	Relative Time
GHS12 (original)(packed) (Gentry et al. 2012)	BGV11 (Brakerski et al. 2011)	AES	Intel Xeon CPU @ 2.0GHz with 256GB RAM	80	40	48 h	54	37 min
GHS12 (original)(byte-sliced) (Gentry et al. 2012)						65 h	720	5 min
CLT13 (byte-wise) (Cheon et al. 2013)	DGHV10 (Van Dijk et al. 2010)	AES	Intel Core i7 @ 3.4GHz with 32GB RAM	72	40	18.3 h	33	33 min
CLT13 (state-wise) (Cheon et al. 2013)						113 h	531	12 min 46 s
CLT14 (state-wise) (Coron et al. 2014)	DGHV10 (Van Dijk et al. 2010)	AES	Intel Xeon E5-2690 @ 2.9GHz	80	40	102 h	1875	3 min 15 s
CLT14 (state-wise) (Coron et al. 2014)				72		3 h 35 min	569	23 s
LN14a (YASHE) (Lepoint and Naehrig 2014)	LTV12 (López-Alt et al. 2012)	SIMON	Intel Core i7-2600 @ 3.4GHz[9]	128	34	1 h 10 min	2,048	2.04 s
LN14a (FV) (Lepoint and Naehrig 2014)	Bra12 (Brakerski 2012)					3 h 27 min	2,048	6.06 s
DHS14 (Doröz et al. 2014)	LTV12 (López-Alt et al. 2012)	AES	Intel Xeon @ 2.9GHz	~80	40	31 h	2,048	55 s
DSES14 (Doröz et al. 2014)	LTV12 (López-Alt et al. 2012)	Prince	Intel Core i7 3770K @ 3.5GHz with 32GB RAM[10]	130	30	57 min	1,024	3.3 s
ARSTZ15 (Albrecht et al. 2015)	BGV11 (Brakerski et al. 2011)	LowMC	Intel Haswell i7-4770K CPU @ 3.5GHz with 16GB RAM	80	12	8 min	600	0.8 s
GHS12 (updated)(no bootstrapping) (Gentry et al. 2012)	BGV11 (Brakerski et al. 2011)	AES	Intel Core i5-3320M at 2.6GHz with 4GB RAM[11]	80	40	4 min 12 s	120	2 s
GHS12 (updated)(with bootstrapping) (Gentry et al. 2012)						17 min 30 s	180	5.8 s

Общедоступные реализации FHE.

Хотя все вышеупомянутые реализации опубликованы в литературе, к сожалению, только некоторые из них общедоступны для исследователей. Некоторые из общедоступных реализаций перечислены в таблице 6. Из общедоступных реализаций HElib (HaleviandShoup 2013b) является наиболее важной и широко используемой. HElib реализует схему BGV (Brakerski et al. 2011) с техникой упаковки зашифрованного текста Smart-Vercauteren и

некоторыми новыми оптимизациями. Дизайн и реализация HElib задокументированы в HaleviandShoup (2013a), а алгоритмы, используемые в HElib, задокументированы в HaleviandShoup (2014). HElib разработан с использованием низкоуровневого программирования, которое имеет дело с аппаратными ограничениями и компонентами компьютера без использования функций и команд языка программирования и, следовательно, определяется как «язык ассемблера для HE». Он был реализован с использованием библиотеки C ++ под лицензией GPL. С декабря 2014 года он поддерживает самозагрузку (HaleviandShoup 2015), а с марта 2015 года поддерживает многопоточность. В важном расширении гомоморфная оценка AES была реализована поверх HElib (Gentryetal. 2012) и включена в исходный код HElib в HaleviandShoup (2013b).

Таблица 6. Некоторые общедоступные реализации FHE

Name	Scheme	Lang	Documentation	Libraries
HElib (Halevi and Shoup 2013b)	BGV (Brakerski et al. 2011)	C++	Yes (Halevi and Shoup 2013a)	NTL, GMP
libScarab (Perl et al. 2011a)	SV (Smart and Vercauteren 2010)	C	Yes (Perl et al. 2011b)	GMP, FLINT, MPFR, MPIR
FHEW (Ducas and Micciancio 2014)	DM14 (Ducas and Micciancio 2015)	C++	Yes (Ducas and Micciancio 2015)	FFTW
TFHE (Chillotti et al. 2017)	CGGI16 (Chillotti et al. 2016)	C++	Yes (Chillotti et al. 2016)	FFTW
SEAL (Laine et al. 2017)	FV12 (Fan and Vercauteren 2012b)	C++	Yes (Chen et al. 2017)	No external dependency

К сожалению, использовать HElib непросто из-за сложности, необходимой для его низкоуровневой реализации и выбора параметров, что влияет как на производительность, так и на уровень безопасности. Еще одна известная реализация FHE с открытым исходным кодом – это libScarab (Perletal. 2011a). Насколько нам известно, libScarab (Perletal. 2011a) - первая реализация FHE с открытым исходным кодом. Его выбор параметров относительно проще, чем у HElib, но он страдает множеством ограничений. Например, он не реализует современные методы (например, методы уменьшения модуля и повторной линеаризации (BrakerskiandVaikuntanathan 2014a)) для управления уровнем шума, а также не поддерживает методы SIMD, представленные в SmartandVercauteren (2014). Он реализует схему FHE SmartVercauteren в SmartandVercauteren (2010), а документация предоставляется в Perl et al. (2011b).

Еще одна важная реализация, представленная Дукасом и Миччанчио, называется «Самое быстрое гомоморфное шифрование на Западе» (FHEW) (DucasandMicciancio, 2014). Это задокументировано в DucasandMicciancio (2015). Это значительно сокращает время, необходимое для начальной загрузки зашифрованного текста, требующего гомоморфной оценки логического элемента И-НЕ, «менее чем за секунду». Шлюз NAND

функционально завершен. Следовательно, любые возможные логические схемы могут быть построены с использованием только вентилей NAND. В DucasandMicciancio (2015) использование упаковки зашифрованного текста и методов SIMD обеспечивает амортизированную стоимость. Однако в FHEW такая производительность достигается с использованием всего нескольких сотен строк кода с использованием одной дополнительной библиотеки, FFTW (FrigoandJohnson 2005). Позже стоимость гомоморфных вычислений любого двоичного логического элемента (DucasandMicciancio, 2015) увеличивается в 50 раз за счет некоторых оптимизаций алгоритма начальной загрузки. Основное улучшение основано на представлении шифров LWE тором. В соответствии с наиболее известным методом начальной загрузки в DucasandMicciancio (2014) стоимость начальной загрузки увеличилась в 10 раз. Они также дополнительно улучшили алгоритмы служебных данных распространения шума с использованием некоторых приближений. Наконец, они также уменьшили размер ключа начальной загрузки с 1 ГБ до 24 МБ, достигнув того же уровня безопасности.

Совсем недавно Microsoft выпустила еще одну библиотеку HE, называемую SimpleEncryptedArithmeticLibrary (SEAL) (Laineetal., 2017). Цель выпуска этой библиотеки объясняется как предоставление хорошо документированной библиотеки HE, которая может быть легко использована как экспертами в области криптографии, так и неспециалистами, не имеющими опыта в области криптографии, такими как практики в области биоинформатики. Библиотека не имеет внешних зависимостей, как другие, и включает в себя инструменты автоматического выбора параметров и оценки шума, что упрощает использование. Наконец, оценки безопасности двух хорошо известных библиотек HE на основе LWE, HElib и SEAL, от атак с двойной решеткой были пересмотрены в Albrecht (2017). Показано, что параметры, обещающие 80-битную безопасность, фактически дают ориентировочную стоимость в 68 бит для SEAL v2.0 и 62 бита для HElib. В заключение мы приводим список универсальных библиотек HE следующим образом: реализация HEAAN, которая поддерживает арифметику с фиксированной точкой (Cheonetal., 2016), библиотеку с ускорением GPU cuHE (Daietal., 2017) и общую решетку криптографическая библиотека PALISADE (Rohloff 2017).

Аппаратные реализации и производство FHE.

Fujitsu Laboratories (2013) объявила о первом известном использовании FHE в производственной среде. Их реализация, о которой сообщается, обеспечивает статистические вычисления и биометрическую аутентификацию с использованием безопасности на основе FHE. Они улучшили FHE за счет пакетирования строковых битов данных. Практическое тестирование этой реализации FHE компанией Fujitsu на момент написания этой статьи еще не завершено. Хотя программные реализации считаются многообещающими для получения практической реализации FHE, все еще существует значительный разрыв между

достигнутой и целевой производительностью. Этот пробел привел к появлению новой альтернативной области исследований в области аппаратных реализаций. Аппаратные решения для ускорения схем FHE и SWHE в основном ориентированы на три платформы реализации: графический процессор (GPU), специализированную интегральную схему (ASIC) и программируемую вентильную матрицу (FPGA) (полезный обзор аппаратных реализаций гомоморфных схем) можно найти в Mooreetal. (2014b)). Хотя графический процессор предназначен для графических целей, его высокопараллельная структура дает большие надежды по сравнению с процессором по эффективности. Следовательно, в некоторых исследованиях предлагается использовать GPU для повышения эффективности гомоморфной оценки (Daietal. 2014; Wangetal. 2014, 2015; Daietal. 2015; Lee et al. 2015). Одним из основных препятствий на пути к практическому FHE является рост шума в операции гомоморфного умножения. Это побудило исследователей найти решение, которое может справиться с большим количеством модульных умножений. Поэтому есть некоторые работы, посвященные именно этой проблеме с использованием специализированных ASIC (Dorözetal. 2013; Wangetal. 2014; Dorözetal. 2015a). Несмотря на потенциал решений GPU и ASIC, большинство предлагаемых исследований основано на реконфигурируемом оборудовании, в частности, на FPGA. Платформы FPGA предлагают не только быстрое преобразование Фурье (FFT), но также некоторые методы оптимизации, такие как теоретико-числовое преобразование (NTT) и быстрое модульное сокращение полиномов на аппаратном уровне. Такая большая и реконфигурируемая среда, предоставляемая ПЛИС, побуждает многих исследователей повысить практичность схем FHE (Казинс и др., 2012; Ван и Хуанг, 2013; Цао и др., 2013; Мур и др., 2013; Чен и др., 2015; Као и др., 2014; Мур и др., 2014a; Казинс и др., 2014; Рой и др., 2015; Пёппельманн и др., 2015; Озтюрк и др., 2015).

В заключение, некоторые реализации SWHE (leveled-FHE) (Gentryetal. 2012) приближаются к приемлемой производительности. Однако методы начальной загрузки в схемах FHE должны быть улучшены, а стоимость гомоморфных умножений должна быть снижена для повышения производительности.

Задание. Шифр Виженера

Шифр Виженера– метод полиалфавитного шифрования буквенного текста с использованием ключевого слова.

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал ДжованБаттистаБеллазо, однако получил имя БлезаВиженера, французского дипломата. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа.

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так

далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabularecta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Например, предположим, что исходный текст имеет вид:

ATTACKATDAWN

Человек, посылающий сообщение, записывает ключевое слово («**LEMON**») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: **ATTACKATDAWN**
Ключ: **LEMONLEMONLE**
Зашифрованный текст: **LXFOPVEFRNHR**

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если буквы A-Z соответствуют числам 0-25, то шифрование Виженера можно записать в виде формулы:

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Перечень основной литературы:

1. Косяков М. С. Введение в распределенные вычисления Электронный ресурс / М. С. Косяков. - Введение в распределенные вычисления. - Санкт-Петербург: Университет ИТМО, 2014. - 155 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397, экземпляров неограничено
2. Ванина М. Ф. Распределенные информационные системы. Технологии реализации распределенных информационных систем: учебное пособие / М. Ф. Ванина, А. Г. Ерохин. - Распределенные информационные системы. Технологии реализации распределенных информационных систем. - Электрон.дан. (1 файл). - Москва: Московский технический университет связи и информатики, 2020. - 132 с. - электронный. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397, экземпляров неограничено
3. Кочеров Ю.Н. Разработка методов и алгоритмов разделения и восстановления данных в модулярных пороговых структурах для распределенных вычислительных сетей Электронный ресурс : монография / Н.И. Червяков / Ю.Н. Кочеров. - Ставрополь : Северо-Кавказский федеральный университет, 2016. - 239 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9296-0865-0, экземпляров неограничено

Перечень дополнительной литературы:

1. Глухов М. М. Алгебра Электронный ресурс / Глухов М. М., Елизаров В. П., Нечаев А. А.: учебник. - 3-е изд., стер. - Санкт-Петербург: Лань, 2020. - 608 с. - Рекомендовано ФГКОУ ВПО «Академия Федеральной службы безопасности Российской Федерации» в качестве учебник для студентов вузов, обучающихся по укрупненной группе направлений подготовки и специальностей «Информационная безопасность». - ISBN 978-5-8114-4775-6, экземпляров неограничено.
2. Грибов А. В., Золотых П. А., Михалёв А. В. Построение алгебраической криптосистемы над квазигрупповым кольцом // Математические вопросы криптографии. – 2010.– Т.1. - № 4. – С. 23–32.
3. Евдокимов М. А. Основы криптологии: учебное пособие / М. А. Евдокимов, Е. А. Райков. - Самара: Самарский государственный технический университет, ЭБС АСВ, 2016. - 88 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/90677.html
4. Высокопроизводительные вычисления и суперкомпьютерная техника : курс лекций: учеб.пособие : Направление подготовки 09.06.01 - Информатика и вычислительная техника. Профиль Математическое моделирование, численные методы и комплексы программ.

Квалификация выпускника Исследователь. Преподаватель-исследователь / сост. П. А. Ляхов ; Сев.-Кав. федер.ун-т. - Ставрополь: СКФУ, 2014. - 49 с. - Неопубликованное издание, экземпляров неограничено

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. http://www.un.org - Сайт ООН Информационно-коммуникационные технологии
2. http://www.intuit.ru – Интернет-Университет Компьютерных технологий.
3. http://github.com/hengxin/awesome-distributed-computing.md. Сайт содержит список ресурсов по распределённым вычислениям и системам.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Методические указания
по организации и проведению самостоятельной работы
по дисциплине
«Математические методы представления данных»
для студентов направления подготовки
02.03.01 Математика и компьютерные науки
направленность (профиль)

«Анализ данных и искусственный интеллект»

Общие положения

Самостоятельная работа - планируемая учебная, учебно-исследовательская, научно-исследовательская работа студентов, выполняемая во внеаудиторное (аудиторное) время по заданию и при методическом руководстве преподавателя, но без его непосредственного участия (при частичном непосредственном участии преподавателя, оставляющем ведущую роль за работой студентов).

Самостоятельная работа студентов (СРС) в ВУЗе является важным видом учебной и научной деятельности студента. Самостоятельная работа студентов играет значительную роль в рейтинговой технологии обучения.

К основным видам самостоятельной работы студентов относятся:

– формирование и усвоение содержания конспекта лекций на базе рекомендованной лектором учебной литературы, включая информационные образовательные ресурсы (электронные учебники, электронные библиотеки и др.);

– написание докладов;

– подготовка к семинарам, практическим и лабораторным работам, их оформление;

– составление аннотированного списка статей из соответствующих журналов по отраслям знаний (педагогических, психологических, методических и др.);

– выполнение учебно-исследовательских работ, проектная деятельность;

– подготовка практических разработок и рекомендаций по решению проблемной ситуации;

– выполнение домашних заданий в виде решения отдельных задач, проведения типовых расчетов, расчетно-компьютерных и индивидуальных работ по отдельным разделам содержания дисциплин и т.д.;

– компьютерный текущий самоконтроль и контроль успеваемости на базе электронных обучающих и аттестующих тестов;

– выполнение курсовых работ (проектов) в рамках дисциплин;

– выполнение выпускной квалификационной работы и др.

Методика организации самостоятельной работы студентов зависит от структуры, характера и особенностей изучаемой дисциплины, объема часов на ее изучение, вида заданий для самостоятельной работы студентов, индивидуальных качеств студентов и условий учебной деятельности.

Процесс организации самостоятельной работы студентов включает в себя следующие этапы:

- подготовительный (определение целей, составление программы, подготовка методического обеспечения, подготовка оборудования);
- основной (реализация программы, использование приемов поиска информации, усвоения, переработки, применения, передачи знаний, фиксирование результатов, самоорганизация процесса работы);
- заключительный (оценка значимости и анализ результатов, их систематизация, оценка эффективности программы и приемов работы, выводы о направлениях оптимизации труда).

Самостоятельная работа по дисциплине **«Математические методы представления данных»** направлена на формирование следующих **компетенций**:

ПК-4.Способен использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ моделирования

ПК-7.Способен подготавливать данные и проводить аналитические работы по исследованию больших данных

1. Цель и задачи самостоятельной работы

Ведущая цель организации и осуществления СРС совпадает с целью обучения студента – формирование соответствующего набора компетенций будущего бакалавра по направлению подготовки «Математика и компьютерные науки».

При организации СРС важным и необходимым условием становятся формирование умения самостоятельной работы для приобретения знаний, навыков и возможности организации учебной и научной деятельности. Целью самостоятельной работы студентов является овладение фундаментальными знаниями, профессиональными умениями и навыками деятельности по профилю, опытом творческой, исследовательской деятельности. Самостоятельная работа студентов способствует развитию самостоятельности, ответственности и организованности, творческого подхода к решению проблем учебного и профессионального уровня.

Задачами СРС являются:

- систематизация и закрепление полученных теоретических знаний и практических умений студентов;
- углубление и расширение теоретических знаний;
- формирование умений использовать нормативную, правовую, справочную документацию и специальную литературу;
- развитие познавательных способностей и активности студентов: творческой инициативы, самостоятельности, ответственности и организованности;
- формирование самостоятельности мышления, способностей к саморазвитию, самосовершенствованию и самореализации;
- развитие исследовательских умений;
- использование материала, собранного и полученного в ходе самостоятельных занятий на семинарах, на практических и лабораторных занятиях, при написании курсовых и выпускной квалификационной работ, для эффективной подготовки к итоговым зачетам и экзаменам.

2. Технологическая карта самостоятельной работы студента

Коды реализуемых компетенций	Вид деятельности студентов	Средства и технологии оценки	Объем часов, в том числе (акад.)		
			СРС	Контактная работа с преподавателем	Всего
7 семестр					
ПК-4	Самостоятельное изучение литературы	Собеседование	16	2	18
ПК-4	Подгготовка отчетов по	Конспект	16	2	18

	лабораторным работам				
ПК-7	Самостоятельное изучение литературы	Собеседование	16	2	18
ПК-7	Подготовка отчетов по лабораторным работам	Конспект	16	2	18
Итого за 7 семестр			**64**	**8**	**72**
Итого			**64**	**8**	**72**

3. Порядок выполнения самостоятельной работы студентом

3.1. Методические рекомендации по работе с учебной литературой

При работе с книгой необходимо подобрать литературу, научиться правильно ее читать, вести записи. Для подбора литературы в библиотеке используются алфавитный и систематический каталоги.

Важно помнить, что рациональные навыки работы с книгой - это всегда большая экономия времени и сил.

Правильный подбор учебников рекомендуется преподавателем, читающим лекционный курс. Необходимая литература может быть также указана в методических разработках по данному курсу.

Изучая материал по учебнику, следует переходить к следующему вопросу только после правильного уяснения предыдущего, описывая на бумаге все выкладки и вычисления (в том числе те, которые в учебнике опущены или на лекции даны для самостоятельного вывода).

При изучении любой дисциплины большую и важную роль играет самостоятельная индивидуальная работа.

Особое внимание следует обратить на определение основных понятий курса. Студент должен подробно разбирать примеры, которые поясняют такие определения, и уметь строить аналогичные примеры самостоятельно. Нужно добиваться точного представления о том, что изучаешь. Полезно составлять опорные конспекты. При изучении материала по учебнику полезно в тетради (на специально отведенных полях) дополнять конспект лекций. Там же следует отмечать вопросы, выделенные студентом для консультации с преподавателем.

Выводы, полученные в результате изучения, рекомендуется в конспекте выделять, чтобы они при перечитывании записей лучше запоминались.

Опыт показывает, что многим студентам помогает составление листа опорных сигналов, содержащего важнейшие и наиболее часто употребляемые формулы и понятия. Такой лист помогает запомнить формулы, основные положения лекции, а также может служить постоянным справочником для студента.

Чтение научного текста является частью познавательной деятельности. Ее цель – извлечение из текста необходимой информации. От того на сколько осознанна читающим собственная внутренняя установка при обращении к печатному слову (найти нужные сведения, усвоить информацию полностью или частично, критически проанализировать материал и т.п.) во многом зависит эффективность осуществляемого действия.

Выделяют ***четыре основные установки в чтении научного текста***:

информационно-поисковый (задача – найти, выделить искомую информацию)

усваивающая (усилия читателя направлены на то, чтобы как можно полнее осознать и запомнить как сами сведения излагаемые автором, так и всю логику его рассуждений)

аналитико-критическая (читатель стремится критически осмыслить материал, проанализировав его, определив свое отношение к нему)

творческая (создает у читателя готовность в том или ином виде – как отправной пункт для своих рассуждений, как образ для действия по аналогии и т.п. – использовать суждения автора, ход его мыслей, результат наблюдения, разработанную методику, дополнить их, подвергнуть новой проверке).

Основные виды систематизированной записи прочитанного:

Аннотирование – предельно краткое связное описание просмотренной или прочитанной книги (статьи), ее содержания, источников, характера и назначения;

Планирование – краткая логическая организация текста, раскрывающая содержание и структуру изучаемого материала;

Тезирование – лаконичное воспроизведение основных утверждений автора без привлечения фактического материала;

Цитирование – дословное выписывание из текста выдержек, извлечений, наиболее существенно отражающих ту или иную мысль автора;

Конспектирование – краткое и последовательное изложение содержания прочитанного.

Конспект – сложный способ изложения содержания книги или статьи в логической последовательности. Конспект аккумулирует в себе предыдущие виды записи, позволяет всесторонне охватить содержание книги, статьи. Поэтому умение составлять план, тезисы, делать выписки и другие записи определяет и технологию составления конспекта.

Методические рекомендации по составлению конспекта:

1. Внимательно прочитайте текст. Уточните в справочной литературе непонятные слова. При записи не забудьте вынести справочные данные на поля конспекта;
2. Выделите главное, составьте план;
3. Кратко сформулируйте основные положения текста, отметьте аргументацию автора;
4. Законспектируйте материал, четко следуя пунктам плана. При конспектировании старайтесь выразить мысль своими словами. Записи следует вести четко, ясно.
5. Грамотно записывайте цитаты. Цитируя, учитывайте лаконичность, значимость мысли.

В тексте конспекта желательно приводить не только тезисные положения, но и их доказательства. При оформлении конспекта необходимо стремиться к емкости каждого предложения. Мысли автора книги следует излагать кратко, заботясь о стиле и выразительности написанного. Число дополнительных элементов конспекта должно быть логически обоснованным, записи должны распределяться в определенной последовательности, отвечающей логической структуре произведения. Для уточнения и дополнения необходимо оставлять поля.

Овладение навыками конспектирования требует от студента целеустремленности, повседневной самостоятельной работы.

3.2. Методические рекомендации по подготовке к практическим и лабораторным занятиям

Для того чтобы практические и лабораторные занятия приносили максимальную пользу, необходимо помнить, что упражнение и решение задач проводятся по вычитанному на лекциях материалу и связаны, как правило, с детальным разбором отдельных вопросов лекционного курса. Следует подчеркнуть, что только после усвоения лекционного материала с определенной точки зрения (а именно с той, с которой он излагается на лекциях) он будет закрепляться на практических занятиях как в результате обсуждения и анализа лекционного материала, так и с помощью решения проблемных ситуаций, задач. При этих условиях студент не только хорошо усвоит материал, но и научится применять его на практике, а также получит дополнительный стимул (и это очень важно) для активной проработки лекции.

При самостоятельном решении задач нужно обосновывать каждый этап решения, исходя из теоретических положений курса. Если студент видит несколько путей решения проблемы (задачи), то нужно сравнить их и выбрать самый рациональный. Полезно до начала вычислений составить краткий план решения проблемы (задачи). Решение проблемных задач или примеров следует излагать подробно, вычисления располагать в строгом порядке, отделяя вспомогательные вычисления от основных. Решения при необходимости нужно сопровождать комментариями, схемами, чертежами и рисунками.

Следует помнить, что решение каждой учебной задачи должно доводиться до окончательного логического ответа, которого требует условие, и по возможности с выводом. Полученный ответ следует проверить способами, вытекающими из существа данной задачи. Полезно также (если возможно) решать несколькими способами и сравнить полученные результаты. Решение задач данного типа нужно продолжать до приобретения твердых навыков в их решении.

3.3. Методические рекомендации по самопроверке знаний

После изучения определенной темы по записям в конспекте и учебнику, а также решения достаточного количества соответствующих задач на практических занятиях и самостоятельно студенту рекомендуется, провести самопроверку усвоенных знаний, ответив на контрольные вопросы по изученной теме.

В случае необходимости нужно еще раз внимательно разобраться в материале.

Иногда недостаточность усвоения того или иного вопроса выясняется только при изучении дальнейшего материала. В этом случае надо вернуться назад и повторить плохо усвоенный материал. Важный критерий усвоения теоретического материала - умение решать задачи или пройти тестирование по пройденному материалу. Однако следует помнить, что правильное решение задачи может получиться в результате применения механически заученных формул без понимания сущности теоретических положений.

УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ СРС

Перечень основной литературы:

1. Косяков М. С. Введение в распределенные вычисления Электронный ресурс / М. С. Косяков. - Введение в распределенные вычисления. - Санкт-Петербург: Университет ИТМО, 2014. - 155 с. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397, экземпляров неограничено
2. Ванина М. Ф. Распределенные информационные системы. Технологии реализации распределенных информационных систем: учебное пособие / М. Ф. Ванина, А. Г. Ерохин. - Распределенные информационные системы. Технологии реализации распределенных информационных систем. - Электрон.дан. (1 файл). - Москва: Московский технический университет связи и информатики, 2020. - 132 с. - электронный. - Книга находится в премиум-версии ЭБС IPR BOOKS. - ISBN 2227-8397, экземпляров неограничено
3. Кочеров Ю.Н. Разработка методов и алгоритмов разделения и восстановления данных в модулярных пороговых структурах для распределенных вычислительных сетей Электронный ресурс : монография / Н.И. Червяков / Ю.Н. Кочеров. - Ставрополь : Северо-Кавказский федеральный университет, 2016. - 239 с. - Книга находится в базовой версии ЭБС IPRbooks. - ISBN 978-5-9296-0865-0, экземпляров неограничено

Перечень дополнительной литературы:

1. Глухов М. М. Алгебра Электронный ресурс / Глухов М. М., Елизаров В. П., Нечаев А. А.: учебник. - 3-е изд., стер. - Санкт-Петербург: Лань, 2020. - 608 с. - Рекомендовано ФГКОУ ВПО «Академия Федеральной службы безопасности Российской Федерации» в качестве учебник для студентов вузов, обучающихся по укрупненной группе направлений подготовки и специальностей «Информационная безопасность». - ISBN 978-5-8114-4775-6, экземпляров неограничено.
2. Грибов А. В., Золотых П. А., Михалёв А. В. Построение алгебраической криптосистемы над квазигрупповым кольцом // Математические вопросы криптографии. – 2010.– Т.1. - № 4. – С. 23–32.
3. Евдокимов М. А. Основы криптологии: учебное пособие / М. А. Евдокимов, Е. А. Райков. - Самара: Самарский государственный технический университет, ЭБС АСВ, 2016. - 88 с. - Текст: электронный // Цифровой образовательный ресурс IPR SMART: [сайт]. - URL: https://www.iprbookshop.ru/90677.html
4. Высокопроизводительные вычисления и суперкомпьютерная техника : курс лекций: учеб.пособие : Направление подготовки 09.06.01 - Информатика и вычислительная техника. Профиль Математическое моделирование, численные методы и комплексы программ. Квалификация выпускника Исследователь. Преподаватель-исследователь / сост. П. А. Ляхов ; Сев.-Кав. федер.ун-т. - Ставрополь: СКФУ, 2014. - 49 с. - Неопубликованное издание, экземпляров неограничено

Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. http://www.un.org - Сайт ООН Информационно-коммуникационные технологии
2. http://www.intuit.ru – Интернет-Университет Компьютерных технологий.
3. http://github.com/hengxin/awesome-distributed-computing.md. Сайт содержит список ресурсов по распределённым вычислениям и системам.