Тема моего диплома – Средства анализа нагрузки на структуры хранения данных.

**(сл2)** На сегодняшний день мало кто будет отрицать значимость информации в жизни человека. А постоянное увеличение скорости и объёмов информации приводит к потребности эффективно хранить и обрабатывать информацию. Для решения этой задачи существует огромное количество различных модулей хранения информации и их комбинаций. Но под каждый конкретный случай необходимо выбрать оптимальный контейнер. Следовательно, возникает необходимость анализа работы контейнеров для выбора оптимального.  
 **(сл3)** На этом слайде можно увидеть примеры необходимости выбора оптимального контейнера. Во-первых, это индекс базы данных. Во-вторых, это системы с ограниченными ресурсами.

**(сл4)** Передо мною была поставлена задача: разработать программное обеспечение, в котором должны быть реализованы:

* 1. Визуализация нагрузки на одном контейнере.
  2. Визуализация сравнения нескольких контейнеров.
  3. Алгоритм получения линейного классификатора для адаптивного выбора оптимального способа хранения данных в зависимости от нагрузки.

**(сл5)** Для понимания проблемы необходимо сказать, что такое нагрузка. Нагрузка представляет собой различные последовательности операций вставки, выборки и удаления. Существует 6 видов нагрузки – они представлены на слайде. Для каждой нагрузки могут быть заданы дополнительные параметры:

1. Количество операций (count). Обязательный параметр для каждой нагрузки. По умолчанию равен 1.
2. Диапазон значений. Минимальное возможное значение сгенерированного ключа;Максимальное значение такого ключа.

**(сл6)** Методом индексации данных может быть как распространенные структуры, как массивы, списки, деревья (сбалансированные бинарные деревья и B-деревья), хеш-таблицы так и сложные контейнеры и их комбинации. Каждый метод индексации данных должен предоставлять базовый интерфейс для работы с контейнером. Такой интерфейс должен содержать три базовых операции:

1. Операция чтения данных из контейнера.
2. Операция вставки данных.
3. Операция удаления данных.

**(сл7)** Для того чтобы оценить какой-либо из методов индексации данных, необходимо понять, что происходит при операциях (запись, выборка и удаление данных) с контейнером.

В наиболее простом случае, внутри контейнера каждую операцию можно разложить на композицию некоторого количества двух простейших операций: сравнения и присваивания, что и показано на слайде 7.

**(сл8,9)** **Общая схема работы программы показана на след слайде.**

1. Берется фиксированное число, равное сумме операций.
2. На основе этого числа строится плоскость, в которой при отсечении положительными осями координат образуется треугольник. Этот треугольник и попадет на конечное изображение.
3. Для каждой точки этого треугольника определяются координаты в трехмерном пространстве IRS (см. пункт 2.6.2).
4. По полученным координатам строится нагрузка, в которой количество каждой базовой операции совпадает с координатой по соответствующей оси (рис. 2.6).
5. Полученная нагрузка применяется к контейнеру (или контейнерам) (см. пункт 2.6.3).
6. На основе полученных значений количества простейших операций определяется цвет пикселя (см. пункт 2.6.4).

**(Сл10)** Для визуализации нагрузки рассматривается плоскость в пространстве IRS. Для удобства расчётов и наглядности строится плоскость так, что сумма координат всегда будет равна константе, т.е. Insert+Remove+Select=const. График преобразов изображен на слайде.

**(сл11).** На этом слайде описан Переход от системы экранных координат к системе IRS с мат выкладками.

**(сл12).** Далее описан способ Определения цвета пикселя для одного контейнера. Для этого используется метод линейного градиента:

* определяется минимальное и максимальное значение среди всех пикселей. Потом получаем коэффициент пикселя по формуле

в итоге цвет определяется по формуле:

**(сл13).** На текущем слайде описан способ Определения цвета пикселя для нескольких контейнеров.

Для каждого контейнера задается свой цвет.

1. Для каждого пикселя ищется минимальное количество операций (по всем тестируемым контейнерам).
2. Выбирается цвет хранилища, соответствующего индексу минимального числа из пункта 2.

**(сл14).** В случае 2х хранилищ необходимо производить классификацию. Классификация – признак, по которому можно сказать, что определенная точка в пространстве IRS лучше подходит для одного контейнера, чем для другого. Простейшим примером классификатора является прямая, по одну сторону от которой оптимальным будет являться один контейнер, а по другую – второй. Для решения задачи был выбран Метод опорных векторов (SVM) с линейным ядром.  
 **(сл15).** На изображении зеленым цветом помечены те точки, для которых оптимальным будет одно хранилище, а красным – другое. Помимо этого на изображении присутствует разделяющая линия, полученная в результате работы алгоритма SVM. При использовании адаптивного хранилища для всех точек выше этой линии будет использоваться один контейнер, для всех ниже – другой.

* **(сл16).** На след слайде представлены Средства реализации
* **(сл17).** А на этом слайде представлены требования к ПО и ап.ср.
* **(сл18).** Программа предоставляет консольный интерфейс изобр на слайде 18.
* **(сл19)**. Результатом работы программы является файл с изображ в формате пнг изображенный на текущем слайде.
* **(сл20)**. Общая схема работы программы изображена на след слайде. Входные данные.Пользовательский ввод. Выходные данные. Сгенерированый png-файл. Уравнение разделяющей плоскости
* (**сл 21**). На этом слайде представлена общая архит программы. Она включает в себя модули…

**(сл 22)** Этот слайд изображает тестирование простого списка.зеленым цветом хорошо, красным плохо. На изображении видно, что контейнер хорошо подходит для большого количества операций поиска и удаления при малом количестве вставок, а также при большом количестве вставок и почти полном отсутствии операций удаления и поиска.

**(сл 23)** Этот слайд изображает тестирование сорт списка. На изображении видно, что контейнер очень хорошо подходит для большого и среднего количества операций поиска и удаления при малом количестве вставок.

**(сл 24)** Этот слайд изображает Тестирование B-дерева. На изображении видно, что контейнер плохо подходит для количества операций вставки примерно равном количеству операций поиска при малом количестве удалений. В остальных случаях контейнер хорошо подходит для различных соотношений операций вставки, удаления и поиска.

**(сл 25)** Этот слайд изображает Тестирование B-дерева. На изображении видно, что контейнер плохо подходит для количества операций вставки примерно равном количеству операций поиска при малом количестве удалений. В остальных случаях контейнер хорошо подходит для различных соотношений операций вставки, удаления и поиска.  
**(сл 26)**. Результаты работы представлены на слайде 26.