**БУ ВО**

**«СУРГУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**Ханты-Мансийского автономного округа – Югры»**

**Политехнический институт**

Кафедра: «Автоматизированные системы обработки информации и управления»

Отчет

по научно-исследовательской работе

«Автоматизированная система определения оптимального времени проведения регламентного обслуживания информационных систем на основе данных технического мониторинга»

Выполнил студент гр. № 606-71м

Бажаев Арман Бейсембаевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Научный руководитель:

старший преподаватель, к.т.н.

Яценко Елена Александровна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Сургут 2019 г.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc535099281)

[1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ СИСТЕМЫ 4](#_Toc535099282)

[2. МАТЕМАТИЧЕСКИЕ СРЕДСТВА АНАЛИЗА И ВЫЧИСЛЕНИЯ ДАННЫХ 8](#_Toc535099283)

[2.1. корреляционный анализ 8](#_Toc535099284)

[2.1. полиномиальная интерполяция 9](#_Toc535099285)

[3. ОБЩИЙ АЛГОРИТМ РАСЧЕТА 12](#_Toc535099286)

[ЗАКЛЮЧЕНИЕ 15](#_Toc535099287)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 16](#_Toc535099288)

[ПРИЛОЖЕНИЕ 1](#_Toc535099289). [Блок-схема алгоритма расчета сроков регламентного обслуживания 17](#_Toc535099290)

[ПРИЛОЖЕНИЕ 2](#_Toc535099291). [Листинг программы расчета значений по интерполяционному полиному 20](#_Toc535099292)

# ВВЕДЕНИЕ

На данный момент для достижения цели создания автоматизированной системы расчетов сроков проведения регламентного обслуживания в предыдущих отчетах были рассмотрены инструментальные средства, позволяющие получить необходимые данные, которые требуются для расчета; проведен анализ систем и выбор периодов сбора данных.

Дальнейшими задачами, поставленными для достижения цели, были сформулированы следующие:

1. проектирование концептуальной модели базы данных, содержащей информацию результатов расчета и статистических данных производительности систем;
2. изучение, выбор и освоение математического аппарата для реализации алгоритма расчета сроков регламентного обслуживания;
3. разработка алгоритма расчета сроков регламентного обслуживания.

В отчете приведено описание промежуточных решений поставленных задачи, приведены блок-схемы алгоритмов.

# 1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ СИСТЕМЫ

Для выполнения цели расчетов сроков регламентного обслуживания требовалось создать модель предметной области, которая позволяла бы с высокой степенью информативности данных проводить вычисления, а также выполнять необходимые плановые аналитические операции.

В результате анализа предметной области была разработана промежуточная инфологическая модель данных (рис. 1):

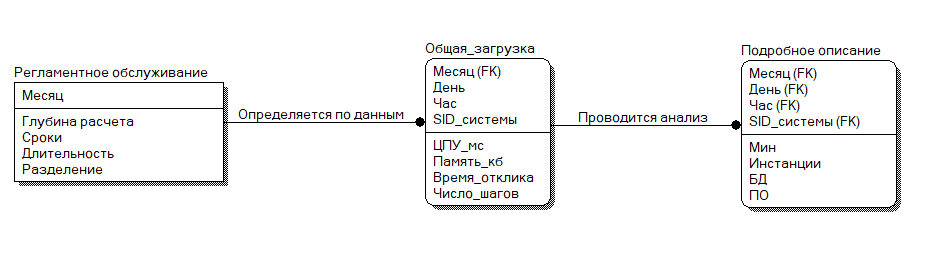


Рис. 1. Инфологическая модель предметной области

Модель состоит из 3 сущностей:

1. «Общая загрузка» («Total\_load»): содержит значения параметров загрузки систем (табл 1. физической модели);

2. «Регламентное обслуживание» («Routine\_maintenance»): содержит данные о результатах расчетов за конкретный месяц с учетом выбранного пользователем режима вычисления, а именно - часы, минуты, (табл. 3 физической модели);

3. «Подробное описание» («Detailed\_description»): содержит статистические данные загрузки систем, являются необходимыми при проверке на критическую загрузку ЦПУ, времени отклика (табл. 2 физической модели).

Физическая БД представлена на рис. 2:

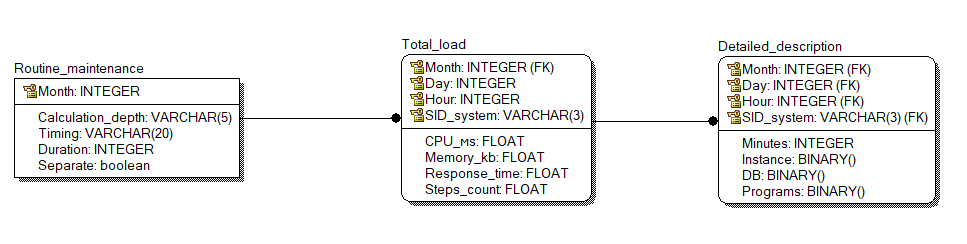
****

Рис. 2. Физическая модель БД

В качестве СУБД для реализации физической модели базы данных была выбрана MySQL.

**Таблица 1**

Общая загрузка

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Значение поля** | **Ограничение целостности** | **Пример значения** |
| Месяц | INTEGER | 1..31 | 1 <= x <= 31 | 1 |
| День | INTEGER | 1…31 | 1 <= x <=31 | 1 |
| Час | INTEGER | 0…23 | 0 <= x <= 23 | 1 |
| SID системы | VARCHAR(5) | Определяется количеством анализируемых систем, 3 символа |  | R3R |
| ЦПУ (мс) | FLOAT | 11111111 | - | 11020202 |
| Память, кб | FLOAT | - | - | - |
| Время отклика, мс | FLOAT | - | - | - |
| Число шагов | INTEGER | - | - | - |

**Таблица 2**

Подробное описание

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | | **Значение поля** | **Ограничение целостности** | **Пример значения** |
| SID системы | VARCHAR(5) | | Определяется количеством анализируемых систем, 3 символа | - | R3R |
| Месяц | | INTEGER | 1…12 | 1 <= x <=12 | 1 |
| День | | INTEGER | 1..31 | 1 <= x <= 31 | 1 |
| Час | | INTEGER | 1..23 | 1 <= x <= 23 | 1 |
| Минута | | INTEGER | 1..59 | 1 <= x <= 59 | 1 |
| Инстанции | | RAW | - | - | - |
| БД | | RAW | - | - | - |
| ПО | | RAW | - | - | - |

**Таблица 3**

Регламентное обслуживание

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Значение поля** | **Ограничение целостности** | **Пример значения** |
| Месяц | INTEGER | 1..12 | 1 <= x <= 12 | 1 |
| Глубина расчета | VARCHAR(5) | Месяц, день | - | Месяц |
| Сроки | VARCHAR(5) | Часовой, минутный период | Начало периода < конец периода | 12-20 |
| Длительность | INTEGER | 1..1000 | - | 25 |
| Разделение | BOOLEAN | True, false | - | True |

Так как период выполнения расчетов в системах имеет циклический характер, максимальное значение которого равно году, при следующем цикле расчетов проводится корректирование данных бинарных файлов, которые хранятся в таблицах, посредством замены соответствующих данных моментов времени.

Хранение файлов, описывающих количество запускаемых транзакций и отчетов, а также загруженностью систем по ЦПУ и памяти необходимо для выполнения проверочных действий на адекватность статистических данных.

Параметры, описывающие детали выполнения регламентного обслуживания (глубина расчета, длительность, целостность) имеют высокую степень значимости при вычислениях, например, шаг рассчитываемой интерполяции, который различен при разных видах расчета.

Размер поля «SID системы» ограничен тремя символами, так как по правилам размер названия идентификатора SAP-системы равен трем буквам.

В таблицах отсутствуют поля вида «Год», так как максимальный размер цикла ведения бизнес-процессов анализируемого предприятия равен одному календарному году. При необходимости использования системы в других организациях требуется модификация логики работы системы, касающаяся не только изменений структуры таблиц базы данных, но и изменения сценариев ведения журналов загрузки систем, которые хранятся в таблице «Детальная загрузка».

# 2. МАТЕМАТИЧЕСКИЕ СРЕДСТВА АНАЛИЗА И ВЫЧИСЛЕНИЯ ДАННЫХ

Для выполнения расчетов необходимы статистические и численные методы, позволяющие выполнить анализ промежуточных расчетных значений и принимать дальнейшие решения относительно полученных результатов.

В результате анализа поставленных в введении задач и имеющемуся набору исходных данных были выбраны следующие вспомогательные математические средства:

1. корреляционный анализ;
2. полиномиальная интерполяция.

## 2.1. корреляционный анализ

Корреляционная связь – связь, описывающая изменение признаков, которые отражают изменение одного признака в соответствии с изменчивостью другого.

Основная задача корреляционного анализа – выявление и оценка связи между случайными величинами [2].

Применение корреляционного анализа обосновано при наличии достаточного количества наблюдений более чем одной переменной.

Мерой корреляции случайных величин служит расчетный коэффициент r. Коэффициент корреляции принимает значения [-1,1]. Значение «-1» описывает отсутствие корреляции между величинами, «1» - полная корреляция величин.

Коэффициент корреляции Пирсона применим для выборки, имеющей нормальное распределение, либо имеет незначительное отклонение. Для порядковых переменных (выборка, имеющее распределение отличающееся от нормального) используется коэффициент корреляции Спирмана или Кендалла.

Выборочный коэффициент корреляции Пирсона «r» рассчитывается по формуле 1:

(1) ,

где , – выборочные средние.

Так как шкала данных производительности систем имеет количественную характеристику в качестве основного коэффициента корреляции был выбран коэффициент Кендалла.

## 2.1. Полиномиальная интерполяция

Интерполирование – поиск промежуточных значений величины по имеющемуся дискретному набору известных значений [1].

Пусть задан набор точек {xi} в некоторой области D и известны значения функции f даны в этих точках {yi}.

Xi – узлы интерполяции, набор узлов – интерполяционная сетка. Разность между соседними узлами интерполяции – шаг интерполяционной сетки.

Основное свойство интерполирующей функции – F(xi) = yi

Примеры интерполяций:

1. линейная интерполяция;
2. интерполяционная формула Ньютона;
3. метод конечных разностей.

Интерполяция функции методом построения многочлена Ньютона основана на разделенных разностях, разделяемых по порядкам. Преимуществом полинома Ньютона является масштабируемость – не требуется выполнять перерасчет при добавлении новых точек.

Количество разделенных разностей определяет точность выходных значений интерполируемых точек.

Рассмотрим распределенные разности некоторых порядков.

Разделенная разность первого порядка – отношение приращения значений функций в узлах (формула 2):

(2)

Разделенная разность второго порядка определяется через разделенные разности первого порядка (формула 3):

(3)

Разделенные разности k-ого порядка определяются через разделенные разности порядка k-1 (формула 4):

(4)

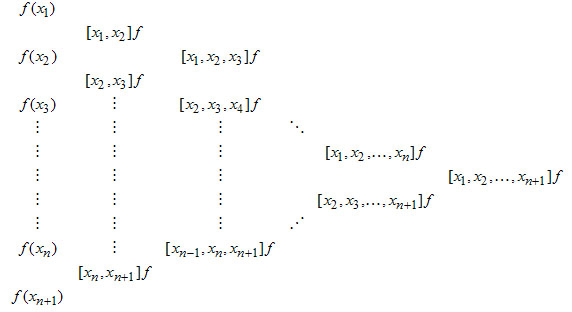
Разделенные разности записывают в виде прямоугольной таблицы (схема Горнера): 

Рис. 3. Схема Горнера

Разделенные разности имеют свойство симметричности, т.е. они не меняют своего значения при перестановке аргументов.

Интерполяционный многочлен Ньютона записывается в следующем виде (формула 5):

(5)

В случае расчета промежуточных значений по исходной выборке становится невозможным рассчитать погрешность измерений, т.к. отсутствует исходная функциональная зависимость. Наименьшая погрешность достигается выборкой небольшого числа точек, между которыми находятся интерполяционные значения, путем деления общего числа исходных данных на части. Листинг программы алгоритма расчета значений представлен в приложении 2.

# 3. ОБЩИЙ АЛГОРИТМ РАСЧЕТА

Алгоритм расчета представлен в приложении 1.

Перед выполнением этапа расчета сроков регламентного обслуживания проводится проверка данных на критические для анализа события. Такие факторы имеют непосредственное влияние на значения выходных показаний, поэтому данный этап является одним из ключевых.

При анализе статистических данных исследуются значения показателей числа шагов, времени отклика системы и базы данных, загрузки ЦПУ. Для нештатной работы системы, которая определяется при обнаружении значений высокой загрузки ЦПУ и времени отклика системы проводятся исследования этих параметров. Среди ситуаций, встречающихся при исследовании значений параметров, возможны такие, что высокая загрузка ЦПУ не является критичной ввиду высокого количества шагов, вызванной большим числом вызовов программ. Противоположным является случай, когда число шагов не противопоставляется числу вызываемых отчетов. Для анализа подобных ситуаций используется инструмент корреляционного анализа, который позволяет определить зависимость числа шагов в системе от выполняемых программ.

Другим критическим событием является высокий показатель времени отклика базы данных системы. Данное значение индивидуально для каждой из систем, поэтому общего правила определения высоких показателей не существует. Основной причиной высокой загрузки систем является высокая загруженность систем отчетности, т.е. системе источнику требуется выполнить запрос к базе данных внешней системы. Ввиду ее загруженности время отклика в исходной системе начинает возрастать, таким образом, достигая пиковых суммарных значений, агенты систем мониторинга фиксируют критический случай и отправляют отчет о высокой нагрузке. Другим фактором, влияющим на высокий показатель времени отклика является выход из строя функциональных узлов системы. Например, рассмотрим кластерные системы, состоящие из двух или более узлов. Применение подобного рода систем обусловлено высокой степенью критичности эксплуатации. Выход из строя одного из узлов кластера способствует перераспределению нагрузки между оставшимися в работоспособном состоянии элементами, таким образом системные показатели начинают резко увеличиваться. Следовательно, анализ показаний, полученных в аварийных состояниях эксплуатации нерентабелен.

Для того чтобы устранить данные нюансы из расчета, выполняется преобразование значений путем замены критических показателей системы на среднее значение по соответствующему моменту времени.

Все исследуемые статистические параметры поделены на 2 группы:

1. критичные (время отклика, ЦПУ);

2. некритичные (число шагов, память).

Анализ времени отклика рассматривается на определение следующих предположений:

1. аварийный случай системы;

2. внеплановые работы в системе.

Исследование осуществляется путем проведения корреляционного анализа между потреблением системой ЦПУ и числом диалоговых сессий в системе.

Высокое потребление ЦПУ и памяти характерно для программ/отчетов имеющих определенные недочеты в их разработке, либо техническими проблемами, связанными с текущей конфигурацией системы.

Анализ ЦПУ и памяти сервера проводится корреляционным анализом с числом диалоговых сессий. В варианте отсутствия корреляции проводится замена значения показаний на среднее по соответствующему числу и дню.

Возможны ситуации, в которых статистические данные указаны в разрезе дня. В таком случае необходимо выполнить поиск значений по каждому из часов. Для этого будет использоваться инструмент интерполяции полиномом Ньютона. Листинг программы расчета приведен в приложении 4, 5. Пример интерполяции – загрузка ЦПУ (рис. 4):

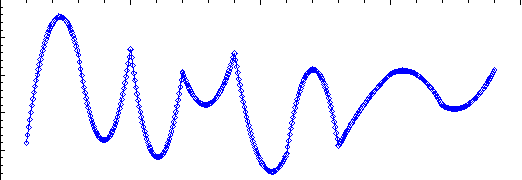


Рис. 4. Интерполяция в разрезе часа

Шаг интерполяции будет выбран в зависимости режима работы системы, т.е. это может быть либо с 08:30 – 17:50, 00:00-23:59 и др. По умолчанию, в разрезе часа шаг интерполяции – 0,041. Аналогично подобному анализу будет выполняться расчет шага при расчете сроков регламентного обслуживания в разрезе минут.

Регламентное обслуживание может быть разделено по часам/минутам. Выбранное условие расчета влияет на конечные расчеты.

# ЗАКЛЮЧЕНИЕ

На данном этапе работы выполнены следующие задачи:

1. спроектирована концептуальная модель базы данных, содержащая информацию результатов расчета и статистических данных производительности систем;
2. проведено изучение, выбор и освоение вспомогательных математических средств реализации алгоритма расчета сроков регламентного обслуживания;
3. разработан алгоритм расчета сроков регламентного обслуживания.

Следующим этапом является дополнение системы корректировкой сроков регламентного обслуживания, путем добавления в систему возможности анализа выполнения разных отчетностей за месяц, и её программная разработка.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

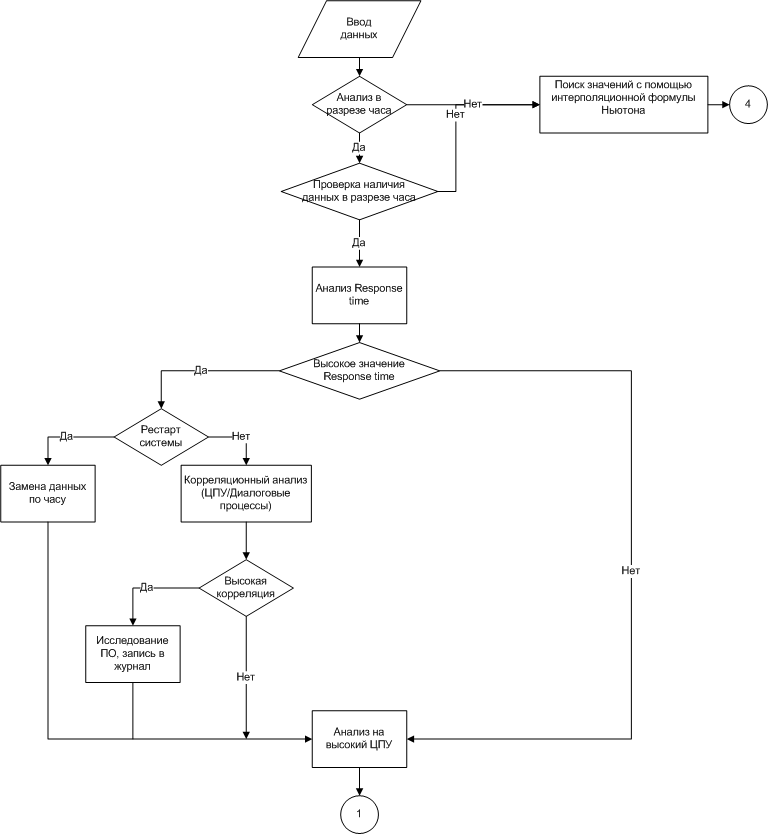
1. Интерполяция [Электронный ресурс] [2019] Режим доступа: https://ru.wikipedia.org/wiki/Интерполяция;

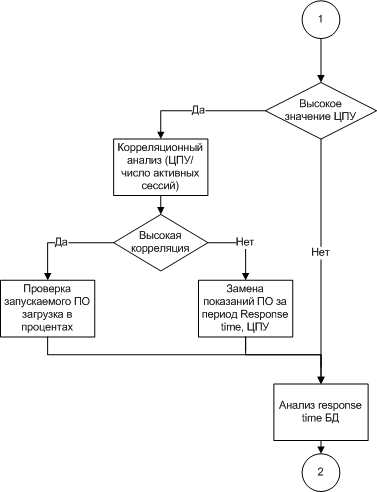
2. Корреляционный анализ [Электронный ресурс] [2019] Режим доступа: https://ru.wikipedia.org/wiki/Корреляционный анализ;

3. Инфологическое моделирование [Электронный ресурс] [2019] Режим доступа: https://www.intuit.ru/studies/courses/1001/297/lecture/7411

# ПРИЛОЖЕНИЕ 1

# Блок-схема алгоритма расчета сроков регламентного обслуживания







# ПРИЛОЖЕНИЕ 2

# Листинг программы расчета значений по интерполяционному полиному

static public double Newton(double x, int n, List<double> MasX, List<double> MasY, double step)

{

double[,] mas = new double[n + 2, n + 1];

for (int i = 0; i < 2; i++)

{

for (int j = 0; j < n + 1; j++)

{

if (i == 0)

mas[i, j] = MasX[j];

else if (i == 1)

mas[i, j] = MasY[j];

}

}

int m = n;

for (int i = 2; i < n + 2; i++)

{

for (int j = 0; j < m; j++)

{

mas[i, j] = mas[i - 1, j + 1] - mas[i - 1, j];

}

m--;

}

double[] dy0 = new double[n + 1];

for (int i = 0; i < n + 1; i++)

{

dy0[i] = mas[i + 1, 0];

}

double res = dy0[0];

double[] xn = new double[n];

xn[0] = x - mas[0, 0];

for (int i = 1; i < n; i++)

{

double ans = xn[i - 1] \* (x - mas[0, i]);

xn[i] = ans;

ans = 0;

}

int m1 = n + 1;

int fact = 1;

for (int i = 1; i < m1; i++)

{

fact = fact \* i;

res = res + (dy0[i] \* xn[i - 1]) / (fact \* Math.Pow(step, i));

}

return res

}

int n = 3;

double step = 7.5

List<double> MasX = new List <double> { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 };

List<double> MasY = new List<double> { 1020000, 1337700, 1253450, 1026900, 1269750, 982600, 1208150, 1122600, 1258350, 972600, 991400, 1214950, 1011650, 1123550, 1202800, 1201500, 1119600, 1122850, 1214650 };

List<double> masY\_temp;

List<double> masX\_temp;

GraphPane pane = zedGraphControl1.GraphPane;

pane.CurveList.Clear();

PointPairList list = new PointPairList();

masX\_temp = new List<double>();

masY\_temp = new List<double>();

for (int X = 1; X <= 17; X += 2)

{

masX\_temp.Clear();

masX\_temp.Add(X);

masX\_temp.Add(X+1);

masX\_temp.Add(X+2);

masY\_temp.Clear();

masY\_temp.Add(MasY[X-1]);

masY\_temp.Add(MasY[X]);

masY\_temp.Add(MasY[X+1]);

for (double X\_temp = masX\_temp[0]; X\_temp <= masX\_temp[2]; X\_temp += 0.0416)

{

list.Add(X\_temp, Newton(X\_temp, 2, masX\_temp, masY\_temp, 1));

}

}

LineItem myCurve = pane.AddCurve("exam", list, Color.Blue, SymbolType.Diamond);

zedGraphControl1.AxisChange();

zedGraphControl1.Invalidate();