МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«**Севастопольский государственный университет**»

кафедра Информационных систем

**Лисянский Александр Игоревич**

Институт информационных технологий и управления в технических системах

курс 4 группа ИCб-42о

09.03.02 Информационные системы (уровень бакалавриата)

**ОТЧЁТ**

по преддипломной практике

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

                                                                Кротов К. В.

(должность) (подпись) (инициалы, фамилия)

Севастополь

2016

Оглавление

[ВВЕДЕНИЕ 3](#_Toc451367446)

[ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ 5](#_Toc451367447)

[Заключение 11](#_Toc451367448)

ВВЕДЕНИЕ

Современные способы повышения производительности выполнения программ при обработке данных связаны с использованием вычислительных кластеров и распределенных вычислительных систем – GRID-систем. Повышение производительности выполнения программ связано с их параллельной реализацией в составе кластера, либо с распределенным выполнением в GRID-системах. Одним из возможных способов повышения производительности выполнения программ обработки данных в составе кластера является их (программ) конвейеризация. Т.е. обработка данных соответствующих им программами осуществляется в составе программного макроконвейера. Конвейеризированное выполнение программ (обработка данных) соответствует классу программных систем МПОД (много программ – одни данные), что предполагает организацию обработки одного потока данных последовательностью программ (последовательностью фрагментов программ) – реализацию программного конвейера. Развитие идей конвейеризации выполнения программ предполагает обработку потоков данных различных типов (в общем случае n типов) соответствующими каждому из этих типов данных программами при реализации последовательного обмена данными между вычислительными сегментами программного конвейера (обмен данными между программами выполняется только после окончания их обработки на сегментах конвейера).

Конвейеризация программ предполагает их разделение на фрагменты, каждый из которых закреплен для выполнения за соответствующим сегментом конвейера (вычислительным устройством, обрабатывающим прибором). Введем в рассмотрение следующие обозначения: i – номер множества однотипных данных, характеризующих одинаковые объекты, которые должны быть обработаны в системе, I – множество всех данных, которые будут обработаны в вычислительной системе, n – количество множеств данных, тогда ),  – количество элементов в множестве однотипных данных, характеризуемых индексом i (множество содержит данные об  одинаковых объектах, в общем виде ). Данные, входящие в некоторое i-е множество, обрабатываются соответствующей им программой. Индекс i соответствует программе, выполняемой в составе конвейера, обрабатывающей данные i-го типа (соответствует типу выполняемой в составе конвейера программы, обрабатывающей данные i-го типа). Однократное выполнение конвейеризированной программы i-го типа обеспечивает обработку одного элемента множества данных i-го типа. Если множество данных i-го типа содержит  элементов, то обрабатывающая эти данные программа должна быть выполнена в конвейерной системе  число раз. Цель функционирования конвейерной системы в этом случае состоит в обработке поступающих на ее вход данных выполняющимися в системе конвейеризированными программами. При этом предполагается, что выполняющие обработку данных конвейеризированные программы находятся в оперативной памяти каждого из сегментов конвейера (загружены в оперативную память вычислительных устройств кластера). Тогда управление вычислительным процессом в конвейерных системах предполагает определение порядка запуска программ обработки данных на выполнение. Т.к. объемы вычислений на каждом сегменте различны, являются различными длительности выполнения программ на соответствующих сегментах, тогда может быть сформировано расписание выполнения конвейеризированных программ обработки соответствующих данных, представляющее собой порядок запуска программ на выполнение.

Постановка задачи управления вычислительным процессом, рассматриваемой в выпускной квалификационной работе, предполагает, что при  () реализуется обработка единичных данных (однократный запуск на выполнение соответствующих программ), для действий с которыми должно быть сформировано расписание выполнения программ. Введем обозначение для интервала времени функционирования системы при обработке данных в виде , где z–индекс временного интервала, тогда , где Z– количество временных интервалов , в течение которых выполняется обработка данных. В этом случае управление вычислительным процессом предполагает формирование совокупности данных, обрабатываемых в течение каждого из временных интервалов и расписаний обработки данных каждой из этих совокупностей. Введем в рассмотрение понятия задания на обработку как совокупности единичных данных, обрабатываемых в течение одного временного интервала . Для задания на обработку введем в рассмотрение обозначение , тогда должно быть сформировано Z заданий (т.е.). Тогда управление вычислительным процессом состоит в формировании эффективных составов заданий  () на обработку данных и расписаний обработки данных этих заданий. Т.к. задачи теории расписаний при количестве приборов более трех и отсутствии каких-либо ограничений на длительности обработки данных являются трудно разрешимыми (NP–трудными), тогда для их решения должны быть применены приближенные методы. Одним их способов формирования приближенных методов теории расписаний является градиентный подход, применение которого к сформулированной задаче будет развито в данной работе.

В результате прохождения преддипломной практики было разработано Windows-приложение, которое строит расписания обработки данных многих типов, с учетом времени перенастройки оборудования с обработки требования одного типа на другой. Учитывая то, что задача составления расписаний является NP-полной, то для её решения использовался метод градиентного поиска, включающий жадную стратегию.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

В ходе прохождения преддипломной практики на основе модели, представленной в выпускной квалификационной работе были сформированы методы приближения к точному решению поставленной задачи (уточнение окрестности поиска решения для заданных начальных значений).

Сформулированные методы были рассмотрены с точки зрения программирования – был написан алгоритм построения решения по составам партий данных для фиксированного типа, алгоритм построения решения по составам партий данных для всех типов данных и алгоритм построения расписания групповой обработки партий данных, полученных с применением вышеупомянутых алгоритмов.

## 2.3 Описание метода формирования составов партий

Метод формирования составов партий данных реализует поиск локально оптимального решения в рамках окрестностей, способ формирования которых формулируется ниже. Для текущего локального оптимального решения по составам партий данных i-ых типов рассматриваются два типа окрестностей. При формировании окрестности первого типа построение решений связано с изменением составов партий данных в рассматриваемом их количестве  (), при формировании окрестности второго типа построение решений связано с увеличением количества  партий данных i-го типа и заданием начального решений для этого количества партий. Таким образом, если построение решений путем изменения составов партий данных -го типа в количестве  является невозможным (выполнены условия окончания формирования составов партий данных -го типа в количестве ), тогда реализуется изменение количества  партий данных этого типа, состав которых будет определяться.

## Алгоритм формирования составов партий всех типов данных

На основе рассмотренного способа формирования решений по составам партий данных некоторого i’-го типа, входящих в окрестность  первого типа для текущего локально оптимального решения , сформулирован алгоритм определения локально оптимального решения по составам партий данных i-ых типов () в соответствии с введенным критерием . Алгоритм определения локально оптимального решения по составам партий данных i-ых типов () содержит следующий порядок шагов:

1) задание для каждого i-го типа данных значений параметров  () равными 1 (– количество решений по составам партий данных i-ых типов, сформированных на текущей итерации алгоритма, хранение которых осуществляется в матрице (соответственно, храниться в матрице ), задается равным 1);

2) формирование начального решения (составов партий данных i-ых типов ()) следующим образом: а)  (); б) инициализация h-ых элементов i-ых строк матрицы (А) на основе выражений вида: (), ; в) проверка условия корректности начального решения по составам партий данных -ых типов ():  и  ; г) если при построении начального решения для какого-либо i-го типа данных () условие его корректности по составам партий не выполняются, тогда для этого типа данных формируется одна фиксированная партия: , , ; при этом формируется новый состав множества I типов данных следующим образом: ; параметр G инициализируется значением 0; инициализация множества :; определение значения  для сформированного начального решения  по составам партий данных (текущего локально оптимального решения);

3) задание -го типа данных, составы партий которого определяются: ;

4) проверка реализации условия , в случае выполнения условия осуществляется инициализация значений параметров , k=0 и элементов матрицы  для i’-го типа данных  (,) (буферизировано решение по составам партий данных i’-го типа, подлежащее оптимизации);

5) определение решений по составам партий данных i’-го типа, формируемых на основе решений, находящихся в матрице , и входящих в окрестность  первого типа текущего локально оптимального решения (для этого вызывается алгоритм формирования составов партий данных i’-го типа, рассмотренный выше, использующий  решений, хранящихся в матрице , для формирования новых решений в окрестности  и определяющий наилучшее решение среди полученных в окрестности );

6) реализация проверки условия  для параметра, полученного из алгоритма формирования решений в окрестности , при  для i’-го типа данных не сформированы новые решения в окрестности , которые могут быть более эффективными, чем рассматриваемое локально оптимальное решение , т.е. не сформирована окрестность  локально оптимального решения, поэтому никакие решения не могут быть проанализированы и на их основе не могут быть сформированы новые решения; выполняется переход на шаг 7; при выполнении условия  осуществляется переход на шаг 14;

7) при  реализуется формирование окрестности второго типа текущего локально оптимального решения (решение в этой окрестности определяется путем увеличения количества  партий данных i’-го типа и формирования начального решения по составам партий для значения ), выполняется модификация количества партий: ; для количества партий данных  выполняется формирование начального состава (формируется решение, входящее в окрестность второго типа рассматриваемого локально оптимального решения): осуществляется инициализация значений ; , , , , элементов матрицы  для i’-го типа данных  (), ;

8) выполняется проверка корректности сформированного начального решения по составам  партий данных: ;  (); при выполнении условия корректности начальных составов  партий реализуется переход на шаг 9; если условие корректности начального решения по составам  партий данных не выполняется, тогда формирование новых решений по составам партий данных i’-го типа является невозможным, текущая i’-ая строка матрицы A соответствует локально оптимальному решению по составам партий данных этого типа; реализуется удаление i’-го типа данных из множества I: ; формирование решений по составам партий данных для i’-го типа данных прекращается, реализуется переход на шаг 14;

9) анализ эффективности начального решения по составам партий данных в количестве  (единственное решение в окрестности второго типа), для этого выполняется инициализация i’-ой строки матрицы А’ решением по составам партий данных, хранимым в строке  матрицы : (); передача сформированного решения  на второй уровень для определения составов групп партий;

10) получение со второго уровня иерархии эффективного решения по составам комплектов, формируемых из данных, обрабатываемых в группах, в виде матрицы , использование решения  для вычисления значения критерия  (значения );

11) для сформированного решения [M(s+g),A’(s+g)] вычисление левого (либо правого) дискретного градиентов  (); вычисление  () выполняется по формулам [2]: а) ; б) ;

12) при выполнении условия  сформированное решение [M(s+g), A’(s+g)] не является более эффективным (при минимизации критерия ), чем решение (для количества партий ), тогда для этого решения выполняется инициализация параметра  (равенство 0 свидетельствует об отсутствии решения в окрестности второго типа, лучшего, чем текущее локально оптимальное решение ); выполняется переход на шаг 14 (при этом единственное сформированное решение в окрестности второго типа сохранено в матрице , параметр );

13) если для сформированного начального решения по составам  партий (строки  матрицы ) выполняется условие , тогда реализуется присваивание  (т.к. это единственное решение в окрестности  второго типа и нет необходимости его сравнения с другими решениями в этой окрестности), а также инициализация ;

14) осуществляется проверка условия ; в случае его выполнения реализуется переход на шаг 3; при  осуществляется переход на шаг 15;

15) проверка условия , при его выполнении отсутствуют типы данных (), для которых сформированные решения по составам партий удовлетворяют условию корректного их состава (т.е. отсутствует возможность определения и анализа новых решений), выполняется переход на шаг 20; при  реализуется инициализация множества  i-ых типов данных, для которых выполняется условие  и множества (); при выполнении условия  на предшествующих итерациях алгоритма сформированы лучшие решения по составам партий данных i-ых типов (), чем текущее локально оптимальное решение ; при  реализуется переход на шаг 16; при  в окрестностях  первого типа либо в окрестностях второго типа локально оптимального решения  нет лучшего решения; выполняется переход на шаг 18;

16) задание -го типа данных, эффективность лучшего решения по составам партий которого сравнивается с эффективностью лучших решений для других i-ых типов (,): ;

17) осуществляется проверка условия ; при его выполнении в окрестности первого типа  либо в окрестности второго типа рассматриваемого локально оптимального решения  не найдено лучшего решения, реализуется переход на шаг 16 (переход к анализу решений для следующего типа данных); в случае выполнения  в окрестности первого либо второго типов получено более эффективное решение по составам партий данных i’-го типа, тогда реализуется сравнение значения  и значений  для лучших решений по составам партий данных i-ых типов (для всех и ); если  (для всех  и ), тогда значение  для лучшего решения по составам партий i’-го типа в окрестности  либо окрестности второго типа для  является максимальным по модулю среди значений градиентов  лучших решений по составам партий данных i-ых типов (и ), параметр  содержит индекс строки матрицы , в которой реализуется хранение решения по составам партий данных i’-го типа, являющегося наиболее эффективным среди решений по составам партий других i-ых типов (); при его выполнении реализуется модификация i’-ой строки в матрице А составов партий при неизменном виде i-ых строк этой матрицы ( и ):  при ; если условие  (для всех  и ) выполняется в случае, когда для текущего локально оптимального решения  рассматривается окрестность второго типа решений по составам партий i’-го типа, тогда реализуется модификация i’-го элемента вектора M; в результате зафиксировано новое локально оптимальное решение  по составам партий данных i-ых типов (); проводится инициализация множества : и переход на шаг 3;

18) идентификация i’-го типа данных, решения по составам партий данных которого буферизируются в матрице : , ; для i’-го типа данных реализуется присваивание:  (, , ), присваивание  (т.е. выполнена буферизация решений по составам партий i’-го типа, находящихся в рассматриваемой окрестности, для последующего формирования на их основе решений, находящихся в окрестности большего размера); в результате сформирована матрица , содержащая решения, на основе которых будут формироваться решения в окрестности  с большей метрикой;

19) проверка выполнения условия , если условие выполняется, тогда реализуется переход на шаг 18; в случае выполнения условия  осуществляется присваивание , k=k+1 и переход на шаг 3;

20) корректное окончание алгоритма.

## Описание алгоритма формирования решения по составам партий заданного типа данных

Для определения локально оптимального решения по составам партий данных i-ых типов () сформулированы: 1) способ формирования решений по составам партий данных некоторого i’-го типа, включаемых в окрестности  первого типа (предполагающий изменение составов партий данных при их количестве ), на основе решений, входящих в окрестность ; 2) обобщенный алгоритм идентификации локально оптимального решения по составам партий данных. Входными параметрами алгоритма формирования решений, включаемых в окрестность  первого типа, на основе решений, входящих в окрестность , являются: идентификатор i’-го типа данных, для которых определяется составы партий; матрица  размерностью , содержащая решения, входящие в окрестность текущего локально оптимального решения; количество партий , составы которых будут изменяться; количество  решений в окрестности ; индекс s текущего шага алгоритма идентификации локально оптимального решения.

Алгоритм формирования решений по составам партий данных i’-го типа в окрестности  на основе решений в окрестности  следующий:

1) инициализация значений параметров , , , , ;

2) инициализация номера партии , состав которой будет изменяться, значением 2 ();

3) модификация значения ; для -ой партии -го типа увеличение количества данных на 1; формирование решения по составам партий (строки  матрицы ): (),,  ; ;

4) проверка выполнения условия  (при заданном количестве партий ); в случае выполнения этого условия сформированное в виде строки  матрицы  решение не интерпретируется (в соответствии с Теоремой 1), тогда ; существует возможность дальнейшего формирования решений, выполняется задание значения параметра j=1 (j–значение шага изменения номера партии h’) и реализуется переход на шаг 5;

5) осуществляется проверка условия ; при  реализуется проверка условия формирования решения по составам партий -го типа в виде  (в соответствии с Теоремой 2); если условие  не выполняется, тогда реализуется переход на шаг 6; при выполнении условия существует возможность формирования дополнительного решения по составам партий данных i’-го типа, тогда , осуществляется переход на шаг 3;

6) в случае выполнения условия  (в соответствии с Теоремой 2) при модификации ()-ой партии будет получено решение, дублирующее полученные ранее; осуществляется модификация параметра j=j+1; выполняется переход на шаг 5;

7) при выполнении  сформированы все возможные решения (хранящиеся в матрице ) с использованием одного решения с индексом  (-ой строки  матрицы ); переход на шаг 8;

8) модификация текущего значения – индекса решения, хранимого в матрице  (осуществляется переход к следующему решению в окрестности , на основании которого могут быть сформированы составы партий данных): , если  (рассмотрены не все решения из окрестности , хранимые в матрице ), тогда реализуется переход на шаг 2; в случае выполнения условия  все решения, хранимые в матрице , находящиеся в окрестности текущего локально оптимального решения, использованы для формирования решений, находящихся в окрестности , хранимых в матрице ; выполняется переход на шаг 9;

9) при  выполняется сравнение решений, хранимых в матрице , с точки зрения дублирования ими друг друга (процедура сравнения предполагает формирование копии матрицы , в копии матрицы  упорядочивание элементов в каждой -ой строке () по убыванию значений элементов, поэлементное сравнение каждой -ой строки () с другими -ыми строками (), удаление из матрицы  строк с индексами , которые соответствуют строкам в копии матрицы , дублирующим рассматриваемую -ю строку в копии ; при удалении q2’-ой строки из матрицы  реализуется изменение ); в результате повторяющиеся решения из матрицы  удалены, остальные решения, хранящиеся в этой матрице (за счет ее дублирования в процедуре сравнения), остались без изменения; инициализация значений  ; ; при  осуществляется переход на шаг 11;

10) реализация проверки условия ; при  не сформированы новые решения в окрестности , которые могут быть более эффективными, чем рассматриваемое локально оптимальное решение A(s), т.е. не сформирована окрестность  локально оптимального решения, выполняется переход на шаг 16;

11) рассмотрение решений по составам партий данных i’-го типа, находящихся в окрестности  текущего локально оптимального решения, с точки зрения их эффективности; для этого выполняется инициализация i’-ой строки матрицы А’ решением по составам партий данных, хранимым в -ой строке  матрицы : (); передача сформированного решения  на второй уровень для определения составов групп партий (при этом исходное локально оптимальное решение A(s) не изменяется);

12) получение со второго уровня иерархии эффективного решения по составам комплектов, формируемых из данных, обрабатываемых в группах, в виде матрицы , использование решения  для вычисления значения критерия  (значения );

13) для сформированного решения [M(s+g),A’(s+g)] вычисление левого (либо правого) дискретного градиентов  (); вычисление  () выполняется по формулам [2]: а) ; б) ;

14) при выполнении условия  сформированное решение [M(s+g), A’(s+g)] не является более эффективным (при минимизации критерия ), чем решение (для количества партий ); осуществляется модификация значения ; если  (т.е. рассмотрены не все решения, находящиеся в окрестности ), тогда осуществляется модификация индекса шага алгоритма g=g+1, реализуется переход на шаг 11 (переход к анализу эффективности следующего решения из окрестности ); если , тогда выполняется переход на шаг 16;

15) если для текущего рассматриваемого решения по составам партий (-ой строки  матрицы ) реализуется , тогда выполняется сравнение значения  с максимальным значением  градиента, достигаемым в рассматриваемой окрестности  (для данных i’-го типа); если , тогда текущее решение по составу партий i’-го типа данных является наилучшим среди рассмотренных решений из окрестности , индекс решения (строки  матрицы )  сохраняется: , модификация значения : ; реализуется переход к следующему решению в окрестности , для этого ; если  (рассмотрены не все сформированные решения, находящиеся в окрестности ), тогда g=g+1 и реализуется переход на шаг 11 (переход к анализу эффективности следующего решения из окрестности ); если  (рассмотрены все решения по составам партий данных i’-го типа, находящиеся в окрестности ), тогда выполняется переход на шаг 16;

16) останов алгоритма.

## Описание построения эффективного расписания обработки партий, вхлдящих в сформированные группы

Алгоритм построения эффективного расписания обработки партий, входящих в группу , предполагает последовательное размещение в последовательностях () всех партии данных i-ых типов () из этой группы и идентификацию эффективного местоположения каждой из этих партий в . Таким образом, алгоритм реализует жадный подход к оптимизации, при котором каждое последующее эффективное решение по размещению одной из партий группы  в последовательностях  формируется на основе эффективных решений по порядку обработки партий, рассмотренных на предыдущих шагах алгоритма. Порядок шагов алгоритма построения эффективного расписания обработки партий рассмотрен с точки зрения размещения в последовательностях () партий данных некоторого i-го типа в количестве  и определения эффективного положения (эффективных позиций) этих партий в последовательностях  (). Алгоритм определения эффективного положения (позиций) партий данных i-го типа в последовательностях  () имеет следующий порядок шагов:

1) некоторое сформированное на предыдущем (s-1)-ом шаге алгоритма решение определяется соответствующим видом матриц  и , номер партии  (номер элемента вектора  ()), данные которой в количестве  размещается в последовательностях ()), инициализируется значением 1 ();

2) в силу того, что – количество партий данных, добавленных в () до s-го шага алгоритма, тогда для партии данных i-го типа определяется индекс  столбца матриц ,, в котором будут размещаться параметры, ей соответствующие: ;

3) добавляемая в последовательности  партия данных i-го типа в количестве  элементов размещается в конце последовательностей (); для этого в i-ой строке -го столбца матриц  и  задаются значения, соответствующие этой партии (элементы , и  инициализируются значениями 1 и  соответственно); модификация матриц  и , связанная с инициализацией элементов -ых столбцов, выполняется следующим образом: ; ; ,  при  и ; сформированное начальное решение по размещению в последовательностях () рассматриваемой партии данных i-го типа фиксируется как локально оптимальное решение: ;

4) значение g индекса шага текущего промежуточного решения инициализируется значением 1 ((s+g) – номер промежуточного шага алгоритма, связанного с определением локального оптимального решения по размещению партии данных i-го типа в количестве  элементов в окрестности текущего решения ), индексу текущего рассматриваемого столбца v матриц  и  присваивается значение ;

5) на основе матриц ,  реализуется вычисление элементов матриц  ;

6) для полученного вида матриц ,  и матриц   определяется значение критерия  и значения элементов матрицы ;

7) в последовательностях  изменяется порядок партий таким образом, что рассматриваемая партия данных i-го типа перемещается на одну позицию в начало ; выполняемые действия с матрицами  и  имеют вид: , , , , , , , , где k – индекс строки в матрицах  и , в которой элементы  и  (предыдущая позиция для рассматриваемой партии, которую она займет на (s+g)-ом шаге алгоритма); в результате формируются модифицированные матрицы , ; индекс v столбца, идентифицирующий местоположение парии в  , модифицируется: ;

8) с использованием матриц ,  вычисляются матрицы и , а также значение критерия;

9) приполучено решение не худшее, чем текущее, тогда решение  фиксируется как локально эффективное: , фиксируется критерия ; значение s=s+g, значение индекса g шага поиска следующего локального оптимального решения в окрестности текущего  задаётся равным 1, выполняется переход к шагу 7;

10) если , тогда проверяется выполнение условия  (– метрика максимально возможной окрестности исходного решения , являющегося локально эффективным, в которой выполняется поиск более эффективного решения ;

11) при выполнения условия  реализуется дальнейший поиск более эффективного решения в окрестности  исходного расписания , модифицируются: индекс g промежуточного шага поиска решения (g=g+1), значение метрики окрестности ; проверяется выполнение условия  (рассматриваемая партия является первой в  () (, )); при истинности условия  выполняется переход к шагу 7, при выполнении условия  выполняется переход к шагу 13;

12) при выполнении условия  реализуется переход к шагу 13;

13) полученное локально эффективное решение фиксируется (фиксируется полученный в результате порядок партий), изменение положения рассматриваемой партии данных i-го типа в количестве  в дальнейшем не реализуется;

14) модификация значений количества партий данных i-го типа, которые должны быть размещены в последовательностях  : ; изменение значения счетчика количества размещенных в   партий данных i-го типа: ; если , тогда реализуется изменение индекса следующей рассматриваемой партии данных i-го типа: ; изменение значения шага алгоритма: s=s+1; изменение номера столбца матриц , , в котором на следующем шаге алгоритма будут размещаться значения параметров, характеризующих добавляемую в   партию данных:; выполнить переход к шагу 3;

15) при выполнении условия  в последовательностях   размещены все партии данных i-го типа, включенные в состав группы партий  в соответствии с решением .

Алгоритм формирования порядка обработки партий позволяет выполнить упорядочивание партий данных каждого i-го типа () в количестве  (в каждой из которых  данных ) в последовательностях  , т.е. определить расписание обработки партий на сегментах конвейера. Реализация алгоритма обработки партий данных должна быть выполнена совместно с реализацией алгоритма формирования составов партий данных и алгоритма определения групп партий, соответствующих их сформированным составам партий.

ЗАКЛЮЧЕНИЕ

В процессе прохождения практики были рассмотрены различные способы формирования партий данных различных требований и различные способы построения расписаний обработки партий данных при наличии ограничений на интервалы обработки.

В результате были получены следующие части разделов ВКР:

* Обоснование методов построения партий данных фиксированного типа и партий данных всех типов;
* Обоснование метода построения комплексного расписания групповой обработки партий данных;
* Алгоритмы построения партий данных фиксированного типа и партий данных всех типов;
* Алгоритм построения комплексного расписания групповой обработки партий данных.

В результате проделанной работы была написана программа формирования комплексного расписания групповой обработки партий данных на основе вышеперечисленных алгоритмов.