## МИЛЕХИНА ТАТЬЯНА ВИКТОРОВНА

Lo

Повышение эффективности кластерных систем обработки информации при решении оптимизационных задач (на примере задачи составления расписания занятий)

Специальность 05.13.01. Системный анализ, управление и обработка информации (в приборостроении)



АВТОРЕФЕРАТ диссертации на соискание ученой степени кандидата технических наук

2 8 ИЮЛ 2011

Москва 2011

Работа выполнена на кафедре Вычислительной техники при институте Московском государственном электронной техники (техническом университете).

Научный руководитель

кандидат технических наук, профессор Лупин Сергей Андреевич.

Официальные оппоненты

доктор технических наук, Шагин Анатолий Васильевич

кандидат, физико-математических наук Шебеко Юрий Александрович

Ведущее предприятие

Институт системного анализа РАН

Защита состоится "ЕТ " сентя фор 2011 года на заседании диссертационного совета Д212.134.02 при Московском государственном институте электронной техники (техническом университете).

124498, Москва, Зеленоград, проезд 4806, МИЭТ.

С диссертацией можно ознакомиться в библиотеке МИЭТ. Автореферат разослан "29" ИКСИЯ 2011 года.

Ученый секретары диссертационного совета

д.т.н., Гуреев А.В.

### Общая характеристика работы

Актуальность проблемы. Системный анализ это развивающийся подход к исследованию сложных объектов и явлений. широко применяемый в различных областях науки. Стремительный переход вычислительных систем на новую параллельную архитектуру порождает проблему их эффективного использования. Приложения. созданные для одноядерных процессоров, могут использовать при работе на современном многоядерном процессоре только одно ядро, т.е. они используют ресурсы системы крайне неэффективно. А если речь илет о работе на кластерных многопроцессорных системах, то положение будет еще сложнее. Выход состоит в том, чтобы создавать программные комплексы, которые учитывают архитектурные особенности кластерных вычислителей, разрабатывать новые и совершенствовать существующие методы и алгоритмы анализа и обработки информации. Такой метод системного анализа как декомпозиция. позволяет рассматривать кластерный вычислитель как совокупность подсистем и оптимизировать работу приложений исходя из их характеристик. Этот подход сможет обеспечить повышение эффективности работы современных кластерных систем при решении актуальных ресурсоемких практических задач в различных областях. Многокритериальные задачи синтеза структуры и определения характеристик систем управления для сложных технических систем, часто также решаются методами оптимизации. С ростом размерности вычислительная сложность алгоритмов оптимизации существенно возрастает, что делает актуальной их реализацию кластерных системах.

Одним из примеров задач многокритериальной оптимизации может служить задача составления расписания занятий в ВУЗе, которая относится к классу вычислительно сложных. Как правило, она решается вручную человеком, имеющим большой опыт работы в конкретном университете. При составлении расписания занятий необходимо учитывать достаточно большое количество критериев, назначаемая степень важности которых оказывает существенное влияние, как на процесс, так и на качество составляемого расписания. составления расписания занятий в ручном режиме занимает несколько недель. Безусловно, полностью повторить работу человеческого мозга при рещении многокритериальных задач пока невозможно. Однако это не означает, что автоматические решения, будут уступать им по качеству. Вычислительные мощности современных компьютеров позволяют за небольшое время синтезировать множество вариантов расписаний, опирающихся на разные показатели.

Актуальность работы определяется выбором в качестве объекта исследований методов и алгоритмов решения задач многокритериальной оптимизации, эффективно использующих вычислительные ресурсы кластерных систем. В качестве примера оптимизационной задачи рассматривается задача составления расписания, которая существенно осложняется включением в новые образовательные стандарты принципа вариативности.

<u>Цель работы и задачи исследования.</u> Работа посвящена исследованию методов повышения эффективности кластерных систем при решении оптимизационных задач за счет адаптации алгоритмов к особенностям их архитектуры.

Для достижения поставленной цели в работе решаются следующие основные задачи:

- 1. Анализ переносимости методов решения задач многокритериальной оптимизации на параллельную платформу.
- 2. Постановка и формализация задачи составления расписания занятий в ВУЗе.
- 3. Разработка критериальных оценок качества решения задачи составления расписания.
- 4. Разработка алгоритма решения задачи составления расписания занятий.
- 5. Анализ методов распределения нагрузки для многопроцессорных систем с распределенной памятью.
- 6. Разработка метода распределения нагрузки для алгоритма оптимизации, повышающего эффективность кластерных вычислителей с распределенной памятью.
- 7. Разработка параллельной программы, адаптированной для кластерных вычислителей с распределенной памятью.
- 8. Экспериментальное исследование эффективности предложенного алгоритма.

<u>Объект и предмет исследования.</u> Объектом исследования являются методы и алгоритмы решения оптимизационных задач.

Предметом исследования являются параллельные алгоритмы многокритериальной оптимизации, повышающие эффективность кластерных вычислителей с распределенной памятью.

Методы исследования. При решении указанных задач были использованы положения теории принятия решений, теории множеств, методы многокритериальной оптимизации и языки параллельного программирования.

<u>Научная новизна.</u> Основной результат диссертационной работы состоит в теоретической разработке и практической реализации нового

подхода рещению задач многокритериальной оптимизации, адаптированного для параллельных вычислительных систем распределенной позволяющего памятью, повысить эффективность вычислений счет минимизации коммуникационных накладных который был опробован на примере рещения составления расписания занятий.

### Положения, выносимые на защиту.

- 1. Результаты анализа переносимости алгоритмов решения задачи составления расписания занятий на параллельную платформу.
- 2. Итерационный алгоритм решения задачи составления расписания.
- Метод распределения нагрузки между узлами кластерной системы для алгоритма решения вычислительной составления расписания, учитывающий архитектурные особенности параллельных вычислительных систем распределенной памятью, обеспечивающий динамическую балансировку загрузки кластера **УЗЛОВ** И повышающий эффективность их использования.
- 4. Параллельная программа автоматического решения задачи составления расписания для многопроцессорных вычислительных систем с распределенной памятью.
- 5. Результаты практического исследования эффективности алгоритма и программы.

Практическая значимость. Предложенный в диссертации подход к построению параллельных программ для решения оптимизационных задач на кластерных системах, обеспечивает повышение их эффективности и может быть использован для широкого круга приложений — логистика, экономическое планирование, управление персоналом и ряд других.

Внедрение результатов. Параллельная программа, разработанная в ходе выполнения диссертации, внедрена в отделе автоматизированных информационных систем МИЭТ. В компании «Разум - технологии» предложенный в диссертации подход к решению оптимизационных задач используется для активного управления трафиком в узлах сети.

Апробация работы. Основные положения диссертационной работы докладывались и обсуждались на Всероссийских межвузовских научнотехнических конференциях студентов и аспирантов «Микроэлектроника и информатика — 2007, 2008», «Актуальные проблемы информатизации. Развитие информационной инфраструктуры, технологий и систем - 2007», IV Международной научно-практической конференции «Современные информационные технологии и ИТ-образование» - 2009, Международной

научно-практической конференции ITEDS'2010 «Информационные технологии. Электронные приборы и системы».

<u>Публикации</u>. По материалам диссертации опубликовано пять тезисов докладов и две статьи, одна из которых в журнале, рекомендуемом ВАК. Получены два свидетельства РФ на программу для ЭВМ.

<u>Структура и объём диссертационной работы</u>. Рукопись диссертационной работы состоит из введения, четырех глав, заключения, списка литературы и приложения. Общий объем работы 136 листов и приложения.

# Содержание работы

Во введении обоснована актуальность, цель работы, ее научная новизна и практическая значимость; представлены применяемые методы исследования, внедрение результатов, апробации и обзор содержания диссертации.

<u>В первой главе</u> работы рассматриваются существующие методы решения задач многокритериальной оптимизации, анализируется их применимость для составления расписания занятий и возможность переноса на параллельную платформу. Показано, что для повышения эффективности кластерных вычислителей при решении оптимизационных задач, необходимо создавать программные комплексы, которые учитывают их архитектурные особенности, разрабатывать новые и совершенствовать существующие методы и алгоритмы анализа и обработки информации.

При составлении расписания учитывается достаточно большое число разнородных критериев и ограничений. При невыполнении отдельных ограничений решение считается некорректным, например, нельзя проводить два занятия в одной группе одновременно. Но есть критерии, например, отсутствие «окон» в расписании, при невыполнении которых задачу можно считать решенной с некоторым показателем качества. Критерии качества решения можно разделить на стандартные, которые учитываются всеми, кто разрабатывал методы и алгоритмы решения данной задачи, и специфические, которые отражают особенности конкретного института. Таким образом, с точки зрения теории принятия решений, задача составления расписания относится к задачам многокритериальной оптимизации. В общем случае, набор критериев является существенно разнородным, что и определяет ее особенность.

Среди методов решения задачи составления расписания можно выделить следующие подходы: метод имитации отжига, алгоритм

раскраски графа, имитационное программирование, линейное целочисленное программирование, генетические алгоритмы.

Большинство предлагаемых методов основываются на некотором «нулевом», начальном варианте расписания занятий, и нацелены на его оптимизацию. При этом итоговый вариант может существенно отличаться от начального. Но «нулевое» расписание необходимо сначала как-то получить. Если в таком качестве использовать решение, полученное быстрыми алгоритмами случайного поиска, то его модификация будет длительным процессом с непредсказуемым результатом. Это вытекает из методов итерационной оптимизации, в которых решение, получаемое на к-ом шаге алгоритма, зависит от к-1 шага. Кроме того, в силу размерности задачи, речь может идти только о поиске локально-оптимальных решений. Не удается использовать в качестве начального приближения и уже разработанное в прошлом ручное расписание, т.к. учебный процесс весьма динамичен и может изменяться от семестра к семестру, и тем более, от курса к курсу: возможно изменение числа факультетов, числа групп студентов, преподавателей и дисциплин.

Таким образом, программный комплекс составления расписания должен качественно генерировать в автоматическом режиме не только окончательное, но и начальное решение задачи.

Как правило, расписание циклично. Занятия повторяются через определенный промежуток времени — неделя или месяц, количество часов для каждой дисциплины ограничено. Это значит, что задачу составления расписания можно сформулировать как задачу линейного целочисленного программирования. При таком подходе интересы субъектов учебного процесса учитываются в виде редуцированных критериев оптимальности.

<u>Во второй главе</u> рассмотрены вопросы формализации задачи составления расписания, выбора критериальных функций, представлена математическая модель.

В общем виде задачу составления расписания можно представить как распределенный по времени процесс выполнения некоторого набора заданий с помощью некоторого набора ресурсов. При этом необходимо учитывать ряд начальных и граничных условий и требований, налагаемых на искомое расписание.

В задаче составления расписания можно выделить следующие объекты:

- студенческие группы (G), имеющие уникальный идентификатор;
- преподаватели (P) или группы преподавателей, проводящие занятие;
- учебные дисциплины (S) с видами и объемом занятий;

## аудитории (R).

Единицей расписания является занятие, представляющее собой совокупность всех перечисленных объектов (каждый тип объекта может иметь не единичное вхождение, например, при потоковом занятии, занятие будет содержать некоторое количество объектов «группа», соответствующее количеству групп в данном потоке).

Так как учебный план формируется заранее, то набор дисциплин для каждой группы уже определен, так же как и объем нагрузки по каждому предмету. В работе использован принцип равномерного распределения всех дисциплин в течение семестра. Таким образом, при известном количестве часов, отведенных для данной дисциплины, известна и периодичность проведения занятия F — еще один атрибут расписания. Зачастую нагрузка преподавателей планируется заранее, и для ряда групп уже назначен преподаватель каждой дисциплины.

В общем виде задачу составления расписания занятий в ВУЗе можно сформулировать в следующем виде — необходимо найти временной интервал T, место проведения занятия R, т.е. соответствующую аудиторию, и преподавателя P. В сформированном расписании у каждого занятия будут определены все атрибуты (G, P, S, R, T, F).

Результат решения задачи синтеза расписания обычно представляется в виде таблицы, это наиболее удобная и понятная форма представления такого рода информации.

Расписание (Time-Table) будем рассматривать как 3-х мерный массив, элемент которого это кортеж:

$$TT_{ijk} = (P, S, R)$$
 , где:  $i = \overline{(1,48)}$  порядковый номер занятия (атрибут  $T$  ),

 $j = \widehat{(1,N)}$  идентификатор группы (атрибут G ),

 $k = \overline{(1,4)}$  порядковый номер недели в месяце (атрибут F ).

При этом элементы таблицы  $TT_{ijk}$  могут быть как различными, так и одинаковыми (если несколько групп объединены в поток).

Это представление расписания для группы, аналогично,  $TT_{ijk} = (G, S, R)$  – для преподавателей, и  $TT_{ijk} = (P, G, S)$  для аудиторий.

Для задачи составления расписания занятий можно выделить «жесткие» ограничения, выполнение которых обязательно для получения корректного расписания, и «мягкие», желательные критерии, которые влияют на качество получаемого расписания.

Любое найденное решение должно удовлетворять ограничениям или пространственным и временным условиям не конфликтности расписания:

- в одной аудитории не может проходить одновременно несколько занятий;
- одно занятие не должно проходить одновременно в разных аудиториях (исключая специальные лабораторные работы, когда может потребоваться несколько аудиторий для проведения занятия);
- у группы не может проходить одновременно несколько разных занятий (исключая дисциплины по выбору);
- преподаватель не может проводить одновременно несколько занятий.

К критериям можно отнести следующие:

- количество «окон» в расписании групп;
- количество «окон» в расписании преподавателей;
- превышение оптимального количества занятий в день у групп;
- превышение оптимального количества занятий в день у преподавателей;
- неэффективная загруженность дня у групп;
- неэффективная загруженность дня у преподавателей;
- пожелания преподавателей;
- ограничение на количество рабочих дней у преподавателей;
- оптимальное количество рабочих дней у преподавателей;
- оптимальное количество дней обучения у групп.

Преподаватели могут предъявлять к расписанию различные пожелания: проведение занятий только в определенные дни или в определенное время, скомпоновать занятия в несколько дней и т.п. Так же могут быть различные пожелания и для занятий: определенная аудитория, в которой его необходимо проводить (характерно для лабораторных работ), проводить лекции до семинаров, временные пожелания.

Кроме того, необходимо учесть и ряд критериев, которые влияют на сходимость процесса формирования расписания. Некоторые занятия имеют преимущество перед другими, например, по количеству участников, т.е. чем больше групп вовлечено в процесс, тем больший приоритет у занятия по данному критерию. Или преимущество по загруженности преподавателя.

В качестве метода оптимизации при решении задачи синтеза расписания был выбран метод линейной свертки, который сводит многокритериальную задачу к однокритериальной. Все частные векторные оценки объединяются в одну.

Выбор метода линейной свертки определяется тем, что все критерии оптимизации разнородны, имеют различные единицы измерения, поэтому применять методы, не сворачивающие множество критериев в один, не имеет большого смысла. На уровне перевода критериев в численную форму определяются границы, в пределах которых данный критерий выполнен удовлетворительно. За выход из допустимых границ назначается штраф. При этом в общем векторном критерии будут учтены все отклонения. Система штрафов приводит нас к тому, что в линейной свертке векторный критерий будет устремлен к минимуму:

$$Sched(x) = \sum_{i=1}^{m} \alpha_i f_i(x) \to \min_{x \in X}; \alpha_i > 0, \sum_{i=1}^{m} \alpha_i = 1.$$

Таким образом, лучшим решением будет то решение, значение Sched(x) которого будет минимально. Дополнительные параметры так же учитываются в данной свёртке и включаются в общую сумму наравне с основными критериями.

Для определения коэффициентов важности в работе использовались два подхода: наборы ненормированных и нормированных критериев. Определение коэффициентов важности  $\alpha_i$  плохо формализованный процесс потому, что сложно выразить все критерии в численном виде и определить отношения предпочтения между ними.

Поскольку занятия конкурируют между собой за ресурсы T и R, для учета условий не конфликтности необходимо решать задачу составления расписания последовательно. То есть на каждом шаге можно поставить только одно занятие из всех возможных, заблокировать используемые ресурсы в соответствующих временных рамках и продолжить расстановку.

С некоторым упрощением алгоритм составления расписания можно представить следующим образом:

- 1. Получение всей необходимой исходной информации: данные о занятиях, о преподавателях, о группах, аудиториях, задание коэффициентов важности  $\alpha_i$ .
- 2. Подсчет штрафной функции для каждого занятия в каждый момент времени.
- 3. Выбор на основе полученных значений целевой функции *Sched(x)* наиболее подходящего занятия, закрепление которого, приводит к минимизации штрафной функции.
- 4. Блокирование ресурсов выбранного занятия в выбранный временной интервал.

 Выполнять шаги 2-5 пока все занятия из списка не будут расставлены, или пока дальнейшая расстановка не стала невозможной.

В такой постановке задачи, подбор весовых коэффициентов осложняется тем, что очень сложно сравнить между собой различные критерии и сопоставить им некоторое числовое значение. Например, для обеспечения сходимости итерационного процесса составления расписания и минимизации критерия нерасставленных занятий, необходимо отдавать предпочтение коллективным занятиям — лекциям. Но при таком подходе возрастает риск формирования окон в расписании отдельных групп. Весовые коэффициенты должны ответить на вопрос, какой из параметров для нас важнее.

Для определения весовых коэффициентов в работе использовался преобразованный метод Саати с использованием так называемой «транзитивной» шкалы. Суть метода заключается в получении цепочки предпочтений ЛПР  $\alpha_{12},\ \alpha_{23},\ \ldots \alpha_{m-l,m}$ . При таком подходе достигается непротиворечивость информации и значительно упрощается диалог с ЛПР. Далее пользуясь соотношением

$$\alpha_{ii} = \alpha_i / \alpha_i$$

условием нормированности вектора α, решается уравнение.

В процессе работы было отмечено, что при определении степеней превосходства все же возникают некоторые противоречия. Избежать этого момента можно путем «двойного диалога» с ЛПР. На первом этапе выстраивается список критериев, убывающий по степени важности. Т.е. ЛПР отвечает на вопрос: «Важнее ли і-й критерий ј-го?». На второй стадии уже определяется степень важности критериев, т.е. «Какое превосходство имеет і-й критерий над ј-м?» (слабое, сильное, очень сильное, абсолютное).

Для того чтобы оценить качество полученного расписания, меру его удовлетворения заданным требованиям, были введены следующие критерии:

- 1. Неравномерность расписания для студентов;
- 2. Неравномерность расписания для преподавателей;
- 3. Превышение максимально допустимого числа занятий в день у студентов;
- 4. Превышение максимально допустимого числа занятий в день у преподавателей;
  - 5. Одно занятие в день у студентов;
  - 6. Одно занятие в день у преподавателей;
  - 7. Совпадение расписания занятий с пожеланиями преподавателей;
  - 8. Доля автоматически расставленных занятий;

- 9. Число преподавателей, у которых не выполнены требования по количеству рабочих дней;
  - 10. Превышение оптимального количества дней занятий у студентов;
- 11. Превышение оптимального количества занятых дней у преподавателей.

Выразим эти критерии через операции с элементами 3-х мерной таблицы расписания, число учебных групп обозначим  $N_{\rm gr}$ , а число преподавателей  $N_{\rm pr}$ :

1) 
$$f_1 = \sum_{g=1}^{Ngr} \sum_{w=1}^{4} \sum_{d=0}^{5} \sum_{p=2}^{7} wGr_{(d^*8+p),g,w}$$
, где  $wGr_{(d^*8+p),g,v} = 1$ , если  $TT_{(d^*8+p),g,w} = \varnothing$ , и  $\exists i \in \overline{(1,p-1)}$  и  $j \in \overline{(p+1,8)}$  такие, что  $TT_{(d^*8+i),g,w} \neq \varnothing$  и  $TT_{(d^*8+i),g,w} \neq \varnothing$ ;

$$f_2 = \sum_{pr=1}^{N_{pr}} \sum_{i=1}^{4} \sum_{d=0}^{5} \sum_{p=2}^{7} w \Pr_{(d^*8+p),pr,w} , \text{ где } w \Pr_{(d^*8+p),pr,w} = 1 , \text{ если } TT_{(d^*8+p),g,w}(,P,) \neq pr \text{ при } 1 \leq g \leq N_{gr} , \exists g_1 \in \overline{(1,N_{gr})} \text{ при } i \in \overline{(1,p-1)}$$
 такое, что  $TT_{(d^*8+i),g_1,w}(,P,) = pr$  и  $\exists g_2 \in \overline{(1,N_{gr})}$  при  $j \in \overline{(p+1,8)}$  такое, что  $TT_{(d^*8+i),g_2,w}(,P,) = pr$  ;

3) 
$$f_3 = \sum_{w=1}^4 \sum_{g=1}^{N_{gr}} \sum_{i=0}^5 mGr_{i,g,w}$$
, где  $mGr_{i,g,w} = (\sum_{m=1}^8 TT'_{(i^*8+m),g,w}) - \max Gr$ , где  $TT'_{(i^*8+m),g,w} = 1$  в том случае, если  $TT_{(i^*8+m),g,w} \neq \emptyset$ ;  $\max Gr$  - максимально допустимое число занятий у группы в день, критерий учитывает только превышение допустимого числа занятий, поэтому суммируются только положительные значения  $mGr_{i,g,w}$ ;

4) 
$$f_4 = \sum_{w=1}^4 \sum_{pr=1}^{N_{pr}} \sum_{i=0}^5 m \Pr_{i,pr,w}$$
, где  $m \Pr_{i,pr,w} = (\sum_{m=1}^8 TT'_{(i*8+m),pr,w}) - \max \Pr$ , где  $TT'_{(i*8+m),pr,w} = 1$ , если  $\exists g \in \overline{(1,N_{gr})}$  такое, что  $TT_{(i*8+m),g,w}(,P,) = pr$ , так $^2$  максимально допустимое число занятий у преподавателя в день, критерий учитывает только превышение допустимого числа занятий поэтому суммируются только положительные значения  $mPr_{i,pr,w}$ ;

$$f_5 = \sum_{\nu=1}^4 \sum_{g=1}^{N_F} \sum_{i=0}^5 oGr_{i,g,\nu} \text{ , где } oGr_{i,g,\nu} = 1 \text{ , если } (\sum_{m=1}^8 TT'_{(i^*8+m),g,\nu}) = 1 \text{ ,}$$
 где  $TT'_{(i^*8+m),g,\nu} = 1$  в том случае, если  $TT_{(i^*8+m),g,\nu} \neq \varnothing$  .

6) 
$$f_6 = \sum_{\nu=1}^4 \sum_{pr=1}^{N_{pr}} \sum_{i=0}^5 o \Pr_{i,pr,\nu}$$
, где  $oPr_{i,pr,\nu} = 1$ , если  $(\sum_{m=1}^8 TT'_{(i \cdot k + m),pr,\nu}) = 1$ , где  $TT'_{(i \cdot k + m),pr,\nu} = 1$ , если  $\exists g \in \overline{(1,N_{gr})}$  такое, что .  $TT_{(i \cdot k + m),g,\nu}(P, P) = pr$ .

7) Этот критерий разделён на два усредненных подкритерия: доля выполненных пожеланий  $f_{71}$  и доля невыполненных пожеланий  $f_{72}$ .

$$f_{7l} = (\sum_{pr=1}^{Npr} (\sum_{w=1}^{4} \sum_{d=0}^{5} \sum_{p=1}^{8} want \Pr_{(d^*8+p),pr,w}) / wantAmount) / Am Pr$$
 , если

 $TT_{(d^*8+p),g,w}(,P,)=pr$  при некотором  $1\leq g\leq N_{gr}$ , и преподаватель заявлял о желании работать в это время, то want  $\Pr_{(d^*8+v),pr,w} = 1$ . WantAmount обшее это количество «положительных» пожеланий для этого преподавателя. Атрг - это общее преподавателей, хишакцакаси «положительные» желания. Однако этот критерий не совсем точно отражает степень выполнения данного критерия, так как количество пожеланий может превышать общую занятость преподавателя. Например, занятость преподавателя - одна пара в неделю, и преподаватель изъявил желание работать, допустим, в понедельник, таким образом, количество «желаемых» занятий будет равно 6 временным отрезкам, а количество реальных - одному. Поэтому в случае, если количество выполненных пожеланий равно общей занятости преподавателя, то считается, что для него этот критерий равен 1.

$$f_{22} = (\sum_{p_{r=1}}^{Npr} (\sum_{w=1}^{4} \sum_{d=0}^{5} \sum_{p=1}^{8} Nwant \Pr_{(d^*8+p), pr, w}) / NwantAmount) / Am \Pr,$$
 ecim

 $TT_{(d^*8+p),g,\nu}(,P,)=pr$  при некотором  $1\leq g\leq N_{gr}$ , а преподаватель просил не ставить в это время занятий, то  $Nwant \Pr_{(d^*8+p),pr,\nu}=1$ . NwantAmount — количество заявлении о не желании преподавать в какое-либо время для данного преподавателя, AmPr — общее количество преподавателей, высказавших «отрицательные» пожелания.

- 8)  $f_8 = (N_{good\_par} / N_{par})$ , где  $N_{par}$  общее количество занятий, .  $N_{good\_par}$  количество расставленных занятий.
- 9)  $f_9 = (\sum_{pr=1}^{Npr} (must\_day_{pr} \sum_{d=0}^{5} Kol\_day_{pr,d}))$ , где  $Kol\_day_{pr,d} = 1$  в том случае, если  $\exists g \in \overline{(1,N_{gr})}$  такое, что  $TT_{(d^*8+m),g,w}(,P,) = pr$  при  $m \in \overline{(1,8)}$  и  $w \in \overline{(1,4)}$ , учитывается только факт превышения количества дней над  $must\_day_{pr}$ .
- 10)  $f_{10} = (\sum_{gr=1}^{Ngr} (opt\_day_{gr} \sum_{d=0}^{5} Kol\_day_{gr,d}))$ , где  $Kol\_day_{gr,d} = 1$  в том случае, если  $\exists i \in \overline{(1,p-1)}$  и  $\exists w \in \overline{(1,4)}$ , что  $TT_{(d^*8+i),gr,v} \neq \emptyset$ ; а  $opt\_day_{gr} = AllStud_{gr} / opt\_stud$ , где  $AllStud_{gr}$  это общее количество занятий для данной группы в неделю, opt\_stud оптимальное количество занятий в день, принятое равным 4.
- 11)  $f_{11} = (\sum_{pr=1}^{Npr} (opt\_day_{pr} \sum_{d=0}^{5} Kol\_day_{pr,d}))$ , где  $Kol\_day_{pr,d} = 1$  в том случае, если  $\exists g \in \overline{(1,N_{gr})}$  такое, что  $TT_{(d^*8+m),g,v}(,P,) = pr$  при  $m \in \overline{(1,8)}$  и  $w \in \overline{(1,4)}$ ; а  $opt\_day_{pr} = AllStud_{pr} / opt\_stud$ , где AllStud\_{pr} это общее количество занятий для данного преподавателя в неделю, opt\_stud оптимальное количество занятий в день, принятое равным 4.

В третьей главе анализируются возможные способы распределения нагрузки между узлами кластерной вычислительной системы с распределенной памятью с целью повышения эффективности их использования при решении задачи, проводится распараллеливание предложенного алгоритма решения задачи составления расписания.

Вычислительная техника развивается очень быстрыми темпами. Мощности современных персональных компьютеров и рабочих станций сопоставимы с производительностью вчерашних суперкомпьютеров. При этом значительные мощности сконцентрированы в относительно небольшом объеме. С практической стороны интерес представляют кластерные вычислительные системы с архитектурой CoPC или CoWS. Их можно реализовать на основе компьютерных классов, которые имеются в любом университете. С точки зрения программирования, такая система относится к параллельным вычислителям с распределенной памятью.

Кластеры CoWS позволяют получить производительность порядка единиц TFlops, что вполне достаточно для многих практических приложений, но обладают слабой системой коммуникации. Это накладывает существенные ограничения на методы построения параллельных приложений и приводит к необходимости использовать средства явной передачи сообщений через коммуникационную среду при межузловом взаимодействии.

Чтобы из последовательной реализации программы сделать код, способный работать на параллельной платформе, необходимо решить три основные запачи:

- выделить в программе ветви вычислений, которые можно исполнять параллельно;
- распределить данные или код по модулям локальной памяти процессоров;
- согласовать распределение данных с параллелизмом вычислений.

Методы распределения нагрузки основаны на репликации и фрагментации данных или кода между узлами параллельной машины. Особенности архитектуры кластера и алгоритма исключают возможность использования фрагментации кода, он должен быть реплицирован на узлах вычислителя. Следовательно, при распределении нагрузки необходимо использовать только манипуляции с данными.

Как было отмечено ранее, процесс составления расписания последователен. На каждом новом шаге решения задачи необходимо знать результаты всех предыдущих шагов, т.е. знать, какие ресурсы уже заблокированы. В противном случае для выполнения условия не конфликтности, на каждом шаге необходимо будет проверять, совместимость решений, принимаемых в этот же момент времени другими узлами. Если обнаружен конфликт, то расчеты оказались напрасны, и на их проведение и на рассылку данных было затрачено лишнее время. К тому же от результатов каждой итерации будет зависеть значение штрафной функции для следующей итерации. Таким образом, на каждом шаге выполнения программы, возможна установка только одного занятия.

Исходя из этих ограничений, предложено реализовывать параллельный алгоритм следующим образом: каждый узел на каждом шаге вычисляет наиболее подходящее занятие для установки на данном этапе, занятие же выбирается из подсписка от общего списка занятий. Подсписок формируется для каждого процессора свой, индивидуальный, подсписки вычислительных узлов не пересекаются. Сформировать такие уникальные списки возможно с использованием ранка процесса.

Предложенный метод основан на разделении данных между узлами, т.е. на фрагментации, а полученный код соответствует SPMD (Single Program Multiple Data) технологии.

В процессе вычислений на очередной итерации может возникнуть ситуация, когда подсписки на некоторых узлах будут уже исчерпаны. Это приводит к тому, что в дальнейших вычислениях эти узлы не будут участвовать и эффективность вычислительной системы снизится. Для устранения подобного явления необходимо выполнять динамическую балансировку загрузки узлов кластера. Для предложенного алгоритма это означает, что периодически необходимо повторно делить список нерасставленных занятий между узлами. Снижение эффективности вычислений может происходить из-за И разнородности вычислительной системы. Для устранения этого эффекта необходимо делить список занятий пропорционально производительности узлов, но это трудно сделать стандартными средствами параллельного программирования. Следует заметить, что в компьютерных классах узлы, как правило, однородны и такая ситуация не является критичной.

Информационное взаимодействие между узлами осуществляется с помощью передачи сообщений (MPI). Алгоритм предполагает полную осведомленность всех узлов на каждом шаге о результатах всех предыдущих шагов каждого узла. Это значит, что обмен сообщениями должен также производиться на каждой итерации, следовательно, необходимо стремиться уменьшить количество пересылок и объем передаваемых данных. Для этого предложено реплицировать исходные данные на каждом узле. Таким образом, узлу достаточно знать лишь порядковый номер занятия, чтобы узнать все его атрибуты. Это позволяет каждому узлу решать задачу поиска локального оптимума в выделенном пространстве опираясь на полную информацию о состоянии всех остальных узлов.

Рассылки осуществляются:

- при выборе наилучшего варианта для установки на данном шаге, узлы обмениваются только данными о штрафной функции;
- при передаче параметров устанавливаемого занятия на данном этапе: порядковый номер занятия, номер аудитории, порядковый номер пары, номер недели месяца.

Обладая полной информацией о занятии, узел может заблокировать необходимые ресурсы, исходя из полученных данных.

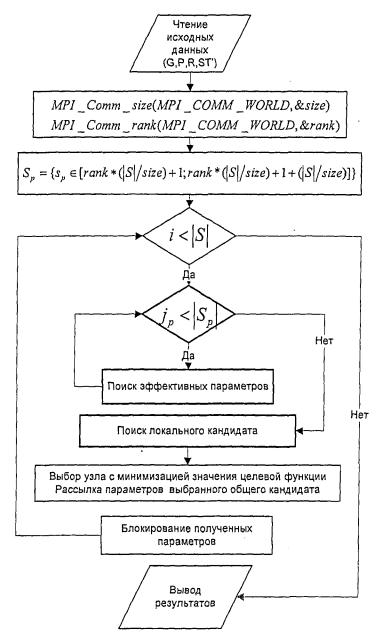


Рис.1. Параллельный алгоритм решения задачи составления расписания.

На рис. 1 представлен параллельный алгоритм задачи составления расписания. Жирными линиями выделены этапы алгоритма, которые исполняются параллельно.

Для практических исследований эффективности предложенного подхода была разработана параллельная MPI программа.

<u>В четвертой главе</u> приведены результаты экспериментальных исследований и испытаний, а также анализ эффективности предложенного алгоритма.

Параллельная программа для решения задачи составления расписания занятий в ВУЗе использует библиотеку МРІ. В качестве аппаратной платформы для экспериментов был использован CoWS кластер МИЭТ-2008 с коммуникационной средой на основе Gigabit Ethernet. Пиковая производительность кластера составляет 1,6 TFlops. Он состоит из 25 узлов, в каждом из которых установлено по два 4-х ядерных процессора Intel Xeon.

Основная цель проводимых исследований заключается в том, чтобы проанализировать возможность эффективного использования параллельных вычислительных систем с распределенной памятью для решения оптимизационной задачи составления расписания.

Некоторые результаты исследований для различного числа занятий приведены на рисунках 2-4.

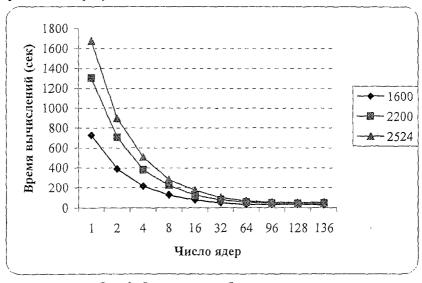


Рис. 2. Оценка масштабируемости алгоритма

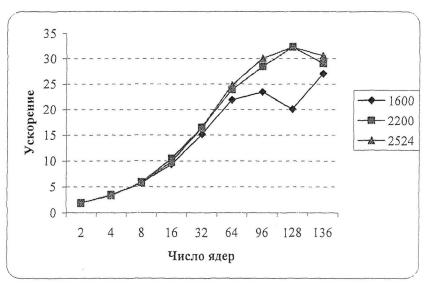


Рис 3. Оценка ускорения параллельного алгоритма

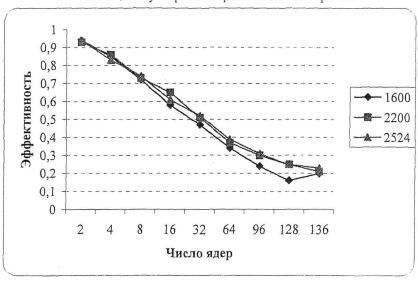


Рис. 4. Эффективность параллельного алгоритма.

Эксперименты проводились для 1600, 2200, 2524 занятий. Результаты показывают, что время вычислений обратно пропорционально

числу вычислительных ядер. Это позволяет сделать вывод о приемлемой масштабируемости алгоритма. При запуске последовательного приложения на 8-ми ядерной машине загруженность процессора составляла величину порядка 12%. Запуск параллельного приложения позволяет повысить эффективность использования ресурсов системы и достичь 90-96% загрузки процессоров узлов кластера, что близко к практическому максимуму и говорит о правильно выбранной стратегии распределения нагрузки.

Проведены исследования влияния различных наборов весовых коэффициентов на качество получаемого решения. Разработанный параллельный алгоритм позволяет автоматизировать процесс поиска близких к оптимальным параметров свертки, обеспечивающих высокое качество получаемых расписаний. В таблице 1 показаны результаты работы алгоритма - значения частных критериев качества, полученные при найденных оптимальных значениях весовых коэффициентов.

Таблица 1. Оценка качества составленного расписания.

a 1. Odenka kajeciba cociabnemore		
Количество занятий		
1600	2200	2524
15	32	29
30	65	188
45	78	182
0	2	34
933	1042	1091
1533	2203	2512
0,68	0,78	0,68
0,02	0,04	0,04
_1	1	1
0	1	2
193	234	262
230	316	362
	Колич 1600 15 30 45 0 933 1533 0,68 0,02 1 0	Количество за   1600 2200   15 32   30 65   45 78   0 2   933 1042   1533 2203   0,68 0,78   0,02 0,04   1 1   0 1   193 234

Отметим, что во всех случаях алгоритм полностью решил задачу, значение критерия  $f_8$  для всех примеров равно 1. Полученные результаты практически подтверждают возможность эффективного использования вычислительных систем с распределенной памятью для решения оптимизационных задач.

В заключении приведены основные результаты работы.

### Основные результаты работы

- 1. Проведен анализ существующих методов решения задач многокритериальной оптимизации и обоснована необходимость их переноса на параллельную платформу.
- 2. Формализована задача составления расписания занятий в ВУЗе, как задача многокритериальной оптимизации.
- 3. Разработан последовательный алгоритм составления расписания занятий в ВУЗе, основанный на минимизации целевой функции, вычисленной по методу линейной свертки.
- 4. Разработан параллельный алгоритм решения задачи составления расписания занятий, учитывающий архитектурные особенности кластерных систем обработки информации, обеспечивающий 90% загрузку узлов кластера и ее динамическую балансировку.
- 5. На языке С++ с использованием библиотеки MPI разработан программный модуль автоматического составления расписания занятий в ВУЗе, адаптированный для кластерных вычислительных систем с распределенной памятью и обеспечивающий 100% расстановку занятий в автоматическом режиме. На указанный программный модуль получено свидетельство об официальной регистрации.
- 6. Исследования предложенного алгоритма и разработанной на его основе программы, проведенные на вычислительном кластере МИЭТ-2008, подтвердили, что эффективность использования узлов кластера в процессе решения оптимизационной задачи достигает величины 90-96%.

## Основные результаты диссертации изложены в работах:

- 1. Милехина Т.В. Формализация задачи составления расписания занятий.// Микроэлектроника и информатика. 14-я Всероссийская межвузовская научно-техническая конференция студентов и аспирантов. М.: МИЭТ, апрель 2007г, с. 157.
- 2. Лупин С.А., Милехина Т.В. Программа автоматического составления расписания занятий В ВУЗе «InstShedule». //Свидетельство РФ об официальной регистрации программы для ЭВМ №2007613815
- 3. Лупин С.А., Милехина Т.В. Метод решения задачи составления расписания, ориентированный на кластерные вычислительные системы. Научно-технический журнал "Известия высших учебных заведений. Электроника" М.: МИЭТ №6, с. 63 69, 2007г.

- Милехина Т.В. Распараллеливание задачи составления расписания занятий в ВУЗе. // «Актуальные проблемы информатизации. Развитие информационной инфраструктуры, технологий и систем». Всероссийская межвузовская научнопрактическая конференция. М.: МИЭТ, ноябрь 2007 г., с. 127.
- 5. Милехина Т.В. Организация межузловых обменов при решении задачи составления расписания. // Микроэлектроника и информатика. 15-я Всероссийская межвузовская научнотехническая конференция студентов и аспирантов. М.: МИЭТ, апрель 2008 г., с. 202.
- 6. Милехина Т.В. Распределение нагрузки и межузловой обмен данными при решении задачи составления расписания занятий на параллельных системах. // «Техника и технология» №5 2008г., стр.41-43.
- 7. Лупин С.А., Милехина Т.В., Ней Мин Тун. Параллельная программа автоматического построения кратчайших связывающих деревьев "ParTrace". //Свидетельство об официальной регистрации программы для ЭВМ № 2008610703
- 8. Милехина Т.В., Лупин С.А., Киселев Д. В. К вопросу об алгоритмизации задачи составления расписания занятий в распределенной среде. // Материалы IV Международной научнопрактической конференции «Современные информационные технологии и ИТ-образование» МГУ, Москва, декабрь 2009г.
- 9. Милехина Т.В., Лупин С.А., Киселев Д. В. Алгоритмизация задачи составления расписания занятий в распределенной среде. // "Материалы Международной научно-практической конференции "Информационные технологии, электронные приборы и системы, ITEDS'2010" 6–7 апреля 2010 г. Минск, Беларусь: БГУ. стр.73-77.

## Подписано в печать:

Заказ № //5 Тираж 90 экз. Уч.-изд.л.1,35. Формат 60х84 1/16.

Отпечатано в типографии МИЭТ (ТУ).

124498, Москва, МИЭТ (ТУ).