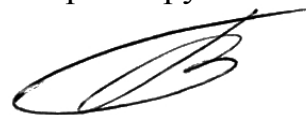


На правах рукописи



Бакулев Александр Валериевич

**МОДЕЛИ И АЛГОРИТМЫ ОРГАНИЗАЦИИ
МОБИЛЬНЫХ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ
В СРЕДЕ МНОГОЯДЕРНЫХ ПРОЦЕССОРОВ**

Специальность 05.13.11 — Математическое и программное
обеспечение вычислительных машин, комплексов и компьютерных сетей

Автореферат
диссертации на соискание ученой степени
кандидата технических наук

Рязань – 2011

Работа выполнена на кафедре систем автоматизированного проектирования вычислительных средств ГОУВПО «Рязанский государственный радиотехнический университет».

Научный руководитель: кандидат технических наук, доцент
Телков Игорь Анатольевич

Официальные оппоненты: доктор технических наук, профессор
Каширин Игорь Юрьевич

кандидат технических наук, доцент
Минаев Юрий Михайлович

Ведущая организация: **Московский государственный институт
электроники и математики (технический
университет)**

Защита состоится 18.02.2011 г. в 12 часов на заседании диссертационного совета Д 212.211.01 в ГОУВПО «Рязанский государственный радиотехнический университет» по адресу: 390005, г. Рязань, ул. Гагарина, д. 59/1.

С диссертацией можно ознакомиться в библиотеке ГОУВПО «Рязанский государственный радиотехнический университет».

Автореферат разослан «12» января 2011 г.

Отзывы на автореферат в двух экземплярах, заверенные печатью организации, просим направлять по адресу: 390005, Рязань, ул. Гагарина, д. 59/1, Рязанский государственный радиотехнический университет.

Ученый секретарь
диссертационного совета
канд. техн. наук, доцент



Пржегорлинский В.Н.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Развитие технологий программирования для персональных компьютеров с точки зрения повышения производительности программ традиционно происходило в основном экстенсивными методами. По большей части разработчики универсального программного обеспечения (ПО) уповали на постоянное совершенствование аппаратных ресурсов компьютера, в частности технологического процесса производства микропроцессоров. Таким образом, очередная установка процессора с увеличенной тактовой частотой позволяла получить ускорение работы прикладных программ без малейшего их изменения.

В настоящее время достигнут фактический предел роста тактовой частоты процессоров и дальнейшее увеличение их производительности становится возможным только за счет перехода к многоядерной архитектуре, по существу к параллельной архитектуре. Так на сегодняшний день для персональных компьютеров уже широко распространены процессоры с 6 и даже 12 ядрами, а в ближайшей перспективе количество ядер может измеряться сотнями. Подобная технологическая революция в архитектуре персонального компьютера предоставляет существенный потенциал для роста производительности ПО, доступный ранее только в мире суперкомпьютеров. Однако для реализации этого потенциала необходимы столь же серьезные революционные изменения современных технологий программирования, а также решение задачи адаптации огромного объема существующего последовательного ПО для эффективного высокопроизводительного выполнения в параллельной вычислительной среде. Термин последовательное ПО подразумевает программы, написанные в расчете на их исполнение на однопроцессорной ВС и не предусматривающие явно организацию параллельных вычислений. В этом смысле без перевода последовательного ПО в параллельную форму выигрыш от использования многоядерных микропроцессоров будет незначительным.

Большое разнообразие существующих сегодня архитектурных платформ, операционных систем (ОС), использующих многоядерные процессоры, а также технологий и языков программирования, положенных в основу последовательного ПО, заставляет искать решение указанной задачи для каждой из них в отдельности. Это создает вторую проблему — проблему переносимости параллельных программ, то есть ставит задачу приведения ПО в соответствие с требованием мобильности. Термин мобильное ПО (portable software) подразумевает программы, которые могут быть откомпилированы и корректно выполнены на вычислительных системах (ВС) с различной архитектурой, без каких-либо изменений.

Сложившаяся ситуация делает важным решение задачи построения и исследования формальных моделей и алгоритмов переноса и распараллеливания универсального последовательного ПО с позиции обеспечения мобильности и эффективности организации его последующего исполнения в среде многоядерных процессоров. Таким образом, тема диссертационной работы является актуальной.

Степень разработанности темы. Научно-методической основой исследований представленных в диссертационной работе являются труды отечественных и зарубежных учёных и специалистов: Глушкова В.М, Воеводина В.В., Воеводина Вл.В., Прангишвили И.В., Виленкина С.Я., Поспелова Д.А., Касьянова В.Н., Корячко В.П., Телкова И.А., Скворцова С.В., Каширина И.Ю., Цейтлина Г.Е., Ющенко Е.Л., Бутомо И.Д., Дейкстры Э., Пратта Т., Таненбаума Э., Дала Э., Хоара Ч.Э., Brandis M.M., Kazi I.H., Steffan J.G., Mowry T.C., Gupta R и других.

В работах Телкова И.А. предложена модель вычислительных алгоритмов и вычислительных систем в рамках единой тензорной методологии, определены подходы к обеспечению мобильности параллельных вычислений, обосновано использование виртуальной параллельной машины, предназначенной для выполнения мобильного ПО на любых типах многопроцессорных ВС. Так как практическая реализация тензорных моделей в силу их большой размерности сопряжена с вычислительными трудностями, в диссертационной работе для формализации концепции параллельной виртуальной машины предложено использовать алгебраический и теоретико-графовый подход.

В работах Глушкова В.М., Цейтлина Г.Е., Ющенко Е.Л. предложен аппарат систем алгоритмических алгебр, предназначенный для исследования свойств алгоритмов и программ. В работах Каширина И.Ю. исследованы свойства алгоритмических языков на базе алгебраических моделей, предложен подход к представлению данных и управляющих конструкций программ на основе формального описания программной машины. Во второй главе диссертации предложено развитие данных подходов с позиции обеспечения мобильности данных, разработано семейство алгебраических систем типов данных.

В работах Корячко В.П. исследованы вопросы построения оптимальных укладок функциональных графов программ применительно к задачам оптимизации числа обращений к оперативной памяти, определены основные понятия и характеристики укладок, предложены методы и алгоритмы укладки графов. В третьей главе диссертационной работы предложено распространить использование методов укладки графов программ на решение задачи синтеза параллельной программы.

В работах Скворцова С.В. предложены модели и методы организации параллельных процессов на мелкозернистом уровне параллелизма на основе теоретико-графового подхода. В диссертационной работе предлагается использовать графовые модели на крупноблочном уровне параллелизма.

В работах Steffan J.G. и Mowry T.C. представлено описание метода спекулятивной многопоточности и проведён анализ возможных способов повышения его эффективности на этапе исполнения, в частности за счет использования аппаратной реализации контроля нарушения зависимостей по данным. В работах Kazi I.H. предложен способ организации спекулятивного многопоточного конвейера для параллельной обработки смежных итераций цикла. В третьей главе диссертационной работы предложены формальная модель метода спекулятивной многопоточности и алгоритмы повышения эффективности его применения на этапе выделения спекулятивных циклических регионов.

Цель диссертационной работы состоит в обеспечении мобильности программ для последующего их высокопроизводительного исполнения в среде многоядерных процессоров путём разработки моделей и алгоритмов организации мобильных параллельных вычислений.

Разработка таких моделей и алгоритмов позволит сократить сроки и дополнительные затраты по переносу последовательных программ в параллельную форму, повысить производительность их исполнения на многоядерных процессорах.

В соответствии с поставленной целью необходимо решить следующие **основные задачи**:

1) произвести анализ и оценку существующих методов и систем распараллеливания последовательных программ, а также способов организации мобильных вычислений;

2) разработать формальные модели мобильного представления последовательных программ;

3) разработать алгоритмы перевода исходных текстов последовательных программ на языках высокого уровня в переносимую параллельную форму;

4) разработать алгоритмы планирования и оптимизации вычислительного процесса исполнения программ, представленных в переносимой параллельной форме, в среде ВС с многоядерными процессорами;

5) спроектировать программные средства, реализующие разработанные модели и алгоритмы, с целью их экспериментальной проверки, апробации и внедрения.

Методы исследования. Для решения поставленных задач в диссертационной работе были использованы следующие методы исследования: теория алгебраических систем, теория графов и математическое программирование, теория расписаний, теория формальных языков.

Научная новизна работы состоит в следующем.

1. Предложена формальная модель мобильного параллельного промежуточного кода (МППК), основанная на алгебраических моделях типов данных и операций языков программирования, обеспечивающая мобильность параллельного представления последовательных программ, заданных на различных языках высокого уровня, отличающаяся использованием нового базиса управляющих операций.

2. Разработаны алгоритмы исключения операций выхода за пределы ветви управляющей конструкции, основанные на предложенной модели МППК, обеспечивающие корректность представления исходных текстов программ. Разработан алгоритм выделения программных регионов, основанный на предложенной модели МППК, обеспечивающий предварительное крупноблочное распараллеливания программы, отличающийся меньшей трудоёмкостью в сравнении с аналогами.

3. Предложена формальная модель подсистемы управления процессами ОС, основанная на предложенной модели МППК и модели автоматной сети, обеспечивающая представление вычислительного процесса на уровне ОС среды многоядерных процессоров.

4. Разработан эвристический алгоритм статического синтеза параллельной программы для мультипроцессорной вычислительной системы с общей памятью, основанный на методе укладки графов, обеспечивающий синтез параллельной программы для ВС с заданным числом процессорных ядер, отличающийся меньшей трудоёмкостью по сравнению с аналогами.

5. Предложена формализация метода спекулятивной многопоточности, используемого для динамического планирования параллельных вычислений, основанная на предложенной модели МППК, обеспечивающая оценку перспективности применения этого метода для распараллеливания циклических участков программы.

6. Разработан алгоритм выделения спекулятивных регионов, основанный на предложенной модели МППК и предложенной формализации метода спекулятивной многопоточности, обеспечивающий повышение эффективности использования данного метода.

Практическая значимость работы состоит в следующем:

- разработаны модели и алгоритмы, позволяющие сократить сроки и дополнительные затраты по переносу последовательных программ в параллельную форму, и повысить производительность их исполнения на многоядерных процессорах;
- спроектированы программы-конверторы исходных текстов последовательных программ с языков высокого уровня *Паскаль* и *Си* в мобильный параллельный промежуточный код;
- разработаны программные средства поддержки этапа исполнения мобильного параллельного промежуточного кода, позволяющие ускорить его выполнение на вычислительной системе с многоядерными процессорами.

Апробация результатов диссертации. Результаты, полученные в рамках работы над диссертацией, докладывались на 5 международных и 8 всероссийских научно-технических конференциях: международных конференциях «Управление и информационные технологии на транспорте» (Санкт-Петербург, 1999), «Проблемы передачи и обработки информации в сетях и системах телекоммуникаций» (Рязань, 1997, 1999, 2000, 2001), всероссийских конференциях «Новые информационные технологии в научных исследованиях и образовании» (Рязань, 1997, 1998, 1999, 2000, 2006, 2008 — 2 доклада, 2009 — 2 доклада, 2010 — 2 доклада).

Публикации. По теме диссертации опубликовано 30 работ: 13 статей (в том числе 3 статьи в журналах из перечня ВАК), 16 тезисов докладов на международных и всероссийских конференциях, одно свидетельство об официальной регистрации программы.

Внедрение результатов работы. Результаты, полученные в диссертационной работе, внедрены в разработки научно-технической продукции филиала ФГУП «ГНПРКЦ «ЦСКБ-Прогресс» - «ОКБ «Спектр» и научно-производственного предприятия ООО «ЭЛЬФ 4М», а также представляют часть НИР № 7-98, № 42/10-01, выполненных в Рязанском государственном радиотехническом университете.

Структура и объем диссертации. Диссертация состоит из введения, четырех глав, заключения, библиографического списка (157 источников). Основной текст работы содержит 177 страниц, 20 таблиц, 22 рисунка. В приложении на 3 страницах приведены акты внедрения результатов и свидетельство о регистрации программы для ЭВМ.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы диссертационной работы, изложены цель и соответствующие ей задачи, приведена структура работы.

В первой главе рассматриваются проблемы организации мобильных параллельных вычислений в среде многоядерных процессоров, анализируются известные языки и системы параллельного программирования, подходы к обеспечению мобильности вычислений, существующие формальные модели последовательных программ, методы и алгоритмы их распараллеливания, делаются выводы и определяются задачи диссертационного исследования.

Выполнен обзор существующих систем параллельного программирования, автоматического и полуавтоматического распараллеливания последовательных программ (DVM-система, mpC, T-Система, V-Ray, OpenMP, Cilk, Argonne/GMD, BERT 77, FORGExplorer, KAP, VAST-C/Parallel, PIPS), в результате которого были отмечены их достоинства и общие недостатки.

Произведён анализ существующих способов организации мобильных вычислений, на основе которого выделено три уровня обеспечения мобильности: уровень языка программирования, уровень ОС, уровень архитектуры ВС и сделан вывод, что в полной мере мобильность можно обеспечить лишь на основе использования концепции промежуточного языка и абстрактной виртуальной машины. В работе рассмотрены реализации программных систем, использующих данную концепцию (BCPL, UCSD Pascal, Java Virtual Machine, Common Language Runtime, Low Level Virtual Machine) и выявлены их общие недостатки.

Проанализированы известные формальные модели данных, программ и алгоритмов (теория схем программ, теоретико-графовые модели, теоретико-множественные модели, машины Тьюринга, нормальные алгорифмы Маркова, программные машины языков программирования, абстрактно-алгебраические модели, системы алгоритмических алгебр, графовые модели, тензорные модели), а также методы и алгоритмы планирования параллельных вычислений (статические и динамические, на этапах анализа и синтеза), были отмечены их достоинства и недостатки.

По результатам выполненного анализа сформулированы выводы и поставлены основные задачи диссертационного исследования.

1. Для обеспечения мобильности вычислений в среде многоядерных процессоров, следует использовать концепцию параллельной виртуальной машины.

2. В рамках данной концепции возникает задача построения оптимального по структуре и набору операций мобильного параллельного промежуточного представления — МППК.

3. Модель МППК должна быть основана на формальных моделях данных и программ языков программирования. При этом выбор был сделан в пользу применения алгебраического и теоретико-графового математического аппарата.

4. Возникает необходимость использования как статических, так и динамических методов на этапе анализа и синтеза параллельных программ на основе МППК, а также их адаптации к особенностям мобильного промежуточного представления путём разработки новых эффективных алгоритмов.

Во второй главе разрабатываются модель мобильного параллельного промежуточного кода и алгоритмы преобразования исходных текстов последовательных программ в мобильный параллельный промежуточный код.

Предложено разбить этап трансляции исходного текста программы в МППК на две стадии. На первой происходит трансляция исходного текста последовательной программы с языка высокого уровня в промежуточное представление. На второй выполняется анализ и предварительное распараллеливание промежуточного представления программы с формированием МППК.

Предложено использовать в качестве промежуточной формы последовательной программы представление, основанное на алгебраической модели типов данных и операций императивных языков программирования высокого уровня. Формальная модель типов данных языков программирования представлена семействами частично универсальных алгебраических систем $\mathfrak{Z}_E = \{T_E^I, T_E^R, T_E^O\}$ и $\mathfrak{Z}_S = \{T_S^A, T_S^S, T_S^U, T_S^R\}$, соответствующих элементарным (тип целых чисел T_E^I , тип действительных чисел T_E^R , перечислимый тип T_E^O) и составным («массив» T_S^A , «структура» T_S^S , «объединение» T_S^U , «ссылочный тип» T_S^R) типам данных. Предложенная модель отличается явной спецификацией атрибутов типов данных, зависящих от машинной реализации и включенных в состав алгебраических систем, что обеспечивает мобильность представления данных программы. Определены основные множества алгебраических систем, задана их сигнатура и приведены примеры представления типов языков программирования Паскаль и Си в предложенной алгебраической модели типов данных.

Для представления операционной семантики конструкций языков программирования предложена алгебраическая модель операций, как совокупность алгебраических систем операций над данными $I_f = \langle O_f, F_f, R_f \rangle$ и управляющих операций $I_c = \langle O_c, F_c, R_c \rangle$. Основное множество O_f составляют допустимые выражения над данными, в качестве базисного подмножества используются операции из сигнатуры алгебраических систем типов данных, представленные в виде функциональных термов. Сигнатура F_f представлена множеством подстановок, позволяющих конструировать производные выражения. Система отношений R_f задаёт приоритеты операций для базисного подмножества.

Важной особенностью алгебраической системы управляющих операций I_c является новый подход к выбору элементов базисного множества, основанный на использовании операционной семантики Дейкстры, выраженной в виде

концепции охраняемых операторов. Предложен следующий состав базиса алгебры управляющих операций: $O_c^0 = \{c^{sq}, c^{gd}, c^{mg}, c^{mc}\}$, где c^{sq} соответствует структуре «последовательность» в аксиоматике Хоара, c^{gd} называется «стражем», задающим условие выполнения вложенного в него оператора, c^{mg}, c^{mc} называются «объединителями» ветвей условной и циклической конструкций соответственно. Множество R_c определяет отношения эквивалентности и вхождения управляющих структур из O_c . Сигнатура алгебры представлена множеством $F_c = \{\otimes\}$ на основе единственной операции композиции управляющих структур. В работе определены правила конструирования производных управляющих операций на основе предложенного базиса. В соответствии с которыми основное множество алгебраической системы было расширено операциями, представляющими основные управляющие конструкции языков программирования высокого уровня $\{c^{if}, c^{for}, c^{wh}, c^{do}, c^{sw}\}$ — «альтернатива», цикл с параметром, цикл с пред и постусловием и «переключатель»:

$$\begin{aligned}
c^{if} &= c_i^{gd}(\rho(o_j^f), c_{i+1}) \otimes c_{i+1} \otimes c_{i+2}^{gd}(\rho(o_j^f), c_{i+3}) \otimes c_{i+3} \otimes c_{i+4}^{mg}(c_i^{gd}, c_{i+2}^{gd}); \\
c^{for} &= c_i^{sq}(f := (x, o_l^f)) \otimes c_{i+1}^{mc}(c_i^{sq}, c_{i+2}^{gd}, c_{i+6}^{gd}) \otimes c_{i+2}^{gd}(\rho(x, o_j^f), c_{i+3}) \otimes \\
&\quad \otimes c_{i+3}(c_{i+4} \otimes c_{i+5}^{sq}(f := (x, o_k^f))) \otimes c_{i+6}^{gd}(\rho(x, o_j^f), \emptyset); \\
c^{wh} &= c_i \otimes c_{i+1}^{mc}(c_i, c_{i+2}^{gd}, c_{i+4}^{gd}) \otimes c_{i+2}^{gd}(\rho(o_j^f), c_{i+3}) \otimes c_{i+3} \otimes c_{i+4}^{gd}(\rho(o_j^f), \emptyset); \\
c^{do} &= c_i \otimes c_{i+1}^{mc}(c_i, c_{i+3}^{gd}, c_{i+4}^{gd}) \otimes c_{i+2} \otimes c_{i+3}^{gd}(\rho(o_j^f), \emptyset) \otimes c_{i+4}^{gd}(\rho(o_j^f), \emptyset); \\
c^{sw} &= c_i^{gd}(\rho(o_j^f), c_{i+1}) \otimes c_{i+1} \otimes \dots \\
&\quad \dots \otimes c_{i+2k}^{gd}(\rho_k(o_j^f), c_{i+2k+1}) \otimes c_{i+2k+1} \otimes \\
&\quad \otimes c_{i+2k+2}^{mg}(c_i^{gd}, c_{i+2}^{gd}, c_{i+4}^{gd}, \dots, c_{i+2k}^{gd}),
\end{aligned}$$

где $\forall c_i \in O_c$, $\forall \rho_k$ — предикат стража, задающий условие выполнения охраняемого оператора, $x \in (E^I \cup E^O)$ — элемент из основного множества алгебры целого или перечислимого типов, $f := \in O_f$, $o_j^f \in O_f$, $o_k^f \in O_f$.

Отличительной особенностью предложенной алгебраической модели операций является возможность представления исходного текста последовательной программы на любом императивном языке программирования в компактной форме с сохранением полной информации об особенностях логической структуры программы, что особенно важно для выявления потенциального параллелизма. Для того, чтобы снять ограничение на использование операторов выхода за пределы ветви управляющей конструкции в произвольной точке *exit*, *break*, *continue*, недопустимых в рамках введенного базиса алгебры управляющих операций, разработан ряд алгоритмов. Все они получают на вход программу, заданную в форме предложенного алгебраического представления, выпол-

няют исключение недопустимых операторов, имеют трудоёмкость, линейно зависящую от числа операций промежуточного представления.

На основе разработанных алгебраических моделей в работе предложена математическая модель МППК в виде системы:

$$PIR = (X, T, V, F, G, \delta_T, \delta_V, \delta_F, \delta_P, \delta_I, \delta_O),$$

где X — множество информационных объектов (переменных и констант), $T \subseteq (T_E \cup T_S)$ — множество типов данных, $V = \{v_i\}$ — множество операторов программы, $F \subseteq (I_f \cup I_c)$ — множество операционных объектов (функций), описываемых алгебраическими системами операций над данными и управляющих операций, $G = \{g_j\}$ — множество регионов программы — подграфов управляющего графа программы, порожденных теми вершинами-операторами, условия выполнения которых одни и те же. Связи между объектами системы PIR заданы следующими отношениями: $\delta_T : X \rightarrow T$ — отображение, определяющее тип информационных объектов; $\delta_V : G \rightarrow V$ — отображение, определяющее распределение операторов по регионам программы; $\delta_F : V \rightarrow F$ — отображение, определяющее функции операторов; $\delta_P : G \rightarrow G$ — отображение, определяющее множество непосредственно вложенных регионов; $\delta_I : V \rightarrow X_I^n \mid X_I \subseteq X$ — отображение, определяющее входную информацию операторов (множество операндов); $\delta_O : V \rightarrow X_O^m \mid X_O \subseteq X$ — отображение, определяющее выходную информацию операторов (множество результатов).

Согласно модели, каждый оператор, входящий в состав j -го региона, рассматривается как тройка $v_i = (\delta_F, \delta_I, \delta_O) \mid v_i \in V_j = \delta_V(g_j) \subseteq V$. Все операторы нумеруются исходя из лексикографического порядка их следования в программе. Программу целиком определяет полный перечень всех операторов $\langle v_1(\delta_F, \delta_I, \delta_O), v_2(\delta_F, \delta_I, \delta_O), \dots, v_m(\delta_F, \delta_I, \delta_O) \rangle$, что отражает операционный подход к представлению программы. Предложена и другая трактовка программы, как совокупности информационных объектов, подвергаемых некоторым операциям. Для этого были заданы отношения δ_R, δ_W , обратные к δ_I, δ_O , а информационные объекты представлены в виде $x_i = (\delta_T, \delta_R, \delta_W)$. Программа тогда представляется как последовательность изменения состояний информационных объектов $\langle x_1(\delta_T, \delta_R, \delta_W), x_2(\delta_T, \delta_R, \delta_W), \dots, x_m(\delta_T, \delta_R, \delta_W) \rangle$.

На основе предложенной модели МППК разработан алгоритм, осуществляющий крупноблочное разбиение программы на совокупность связанных регионов. Предложенное алгебраическое представление программы позволило осуществить построение графа регионов за один просмотр списка управляющих конструкций программы, с трудоёмкостью алгоритма порядка $O(n)$ от числа операций МППК, аналогичные известные алгоритмы имеют сложность порядка $O(n^3)$.

Выявление потенциального параллелизма между регионами МППК основывается на анализе зависимостей составляющих частей программы друг от друга. В работе определены типы отношений между регионами с позиции их параллельного исполнения, а также необходимые условия их существования. Поскольку на этапе трансляции не всегда удаётся выявить полностью все информационные зависимости между операторами программы, существование возможного нарушения информационной зависимости, предложено выразить в явном виде отношениями δ_{out-in}^{∞} — возможная истинная зависимость, δ_{in-out}^{∞} — возможная антивисимость, $\delta_{out-out}^{\infty}$ — возможная выходная зависимость, благодаря чему на этапе исполнения МППК становится возможным окончательно установить факт наличия или отсутствия такой зависимости.

В третьей главе разработаны модели и алгоритмы синтеза параллельных программ на основе модели МППК, предложенной во второй главе диссертации, ориентированные на организацию исполнения МППК в среде целевой ВС с многоядерными процессорами и использование крупноблочного параллелизма на уровне задач.

Так как в среде исполнения универсальных ВС с многоядерными процессорами за планирование вычислений отвечают сервисы ОС, в работе разработан иерархическая модель подсистемы управления процессами ОС, выполненная по результатам анализа наиболее общих закономерностей современных многозадачных ОС. На верхнем уровне предложено использовать алгебраическую модель подсистемы управления процессами $M = \langle \Theta, \varpi \rangle$, средний уровень представляет сеть управляющих автоматов $\mathcal{R}(\Gamma)$, а на нижнем уровне используется предложенная во второй главе модель МППК.

В состав основного множества $\Theta = \{Z, P, I, \Omega, \Lambda, \Pi, \Delta, \Gamma\}$ алгебраической модели входят соответственно: множество задач, процессов, информационных объектов, объектов синхронизации, управляющих автоматов, уровней приоритетов процессов, система диспетчеризации процессов, граф информационных связей между процессами. Сигнатура модели $\varpi = (O, \Phi)$ задаёт множество отношений $O = \{\varepsilon, \rho, \xi, \eta, \sigma\}$ на объектах модели и множество функций $\Phi = \{\phi_{create}, \phi_{terminate}, \phi_{priority}\}$ на элементах множества процессов. Определены следующие отношения: ε описывает состав задачи $z_i \in Z$, как совокупность её процессов $z_j \in P_i$, $\varepsilon: Z \rightarrow P \mid \exists(z_i \varepsilon z_j) \Rightarrow z_i \in Z, z_j \in P_i, P_i \subseteq P, \forall z_j \in z_i$; ρ описывает структуру управляющих связей процесса вычисления и разделяет множество процессов P на подмножества родительских $P^A \subseteq P$ и дочерних $P^D \subseteq P$, $\rho: P^A \rightarrow P^D \mid \exists(z_i \rho z_j) \Rightarrow z_i \in P^A, z_j \in P^D, P^A \subseteq P, P^D \subset P, i \neq j$; $\xi = (P \cup I, \Gamma)$ задает структуру информационного графа $\Gamma \subseteq (P \times I) \cup (I \times P)$, отражающего всю совокупность информационных зависимостей процессов задачи; $\eta: P \rightarrow \Pi$ взвешивает элементы множества процессов значениями приоритетов обслуживания; $\sigma: I \rightarrow \Omega$ определяет закрепление объектов синхро-

низации за разделяемыми между конкурирующими процессами информационными объектами.

Так как вычислительный процесс носит асинхронный характер, в рамках среднего уровня иерархической модели предложено представить его в виде сети $\mathfrak{R}(\Gamma)$ взаимосвязанных управляющих автоматов двух классов. Отдельный автомат класса $A = \langle \alpha, \varphi, S, \delta, \lambda, s_1 \rangle$ однозначно сопоставляется каждому элементу из множества процессов $z_i \rightarrow A_i \mid \forall z_i \in P, A_i \in A \subseteq \Lambda, i = \overline{1, n}$, отдельный автомат $B = \langle \beta, \Psi, \Sigma, \delta', \lambda', \varepsilon_1 \rangle$ — соответственно каждому элементу, принадлежащему множеству информационных объектов $\gamma_j \rightarrow B_j \mid \forall \gamma_j \in I, B_j \in B \subseteq \Lambda, j = \overline{1, m}$. Информационный граф Γ определяет связи — линии обмена между автоматами разных типов составляющих сеть $\mathfrak{R}(\Gamma)$. При этом смена состояния автомата каждого типа приводит к выработке определенного набора выходных сигналов, на основе которых происходит срабатывание связанных с ним автоматов другого типа. В работе определены входной и выходной алфавиты двух классов автоматов, заданы функции их входов и выходов.

На основе предложенной во второй главе модели МППК и модели подсистемы управления процессами разработан алгоритм статического синтеза параллельной реализации последовательной программы. Процесс синтеза рассматривается как задача построения укладки $\pi(A)$ макрографа программы $A = (G, U, \mathfrak{Z})$, заданного на множестве вершин-регионов МППК $g_i \in G, i = \overline{1, N}$, взвешенных временем их выполнения $\tau(g_i) \in \mathfrak{Z}$ на одном процессорном элементе в тактах процессора, связанных дугами U информационных и управляющих зависимостей, в пространственно-временную L^2 -решётку PT , образованную парой ортогональных осей координат, с осью P пространства процессорных элементов ВС и осью T времени выполнения в тактах процессора.

В работе введён ряд характеристик укладки: проекция вершины графа $\pi(g_i) = (g_i^p, g_i^t) \mid g_i \in G, i = \overline{1, N}$, определяющая процессорный элемент, исполняющий программный регион в заданный момент времени; t_s -сечение укладки $G(t_s) = \{g_i \mid g_i^t = t_s, g_i \in G\}$, определяющее множество параллельно исполняемых регионов в момент времени t_s ; p_k -сечение укладки $G(p_k) = \{g_i \mid g_i^p = p_k, g_i \in G\}$, определяющее множество регионов, назначенных на исполнения на заданный процессорный элемент p_k ; ширина t_s -сечения укладки $W_s(\pi(A)) = |G(t_s)|$; общая ширина укладки $W(\pi(A)) = \max_{s=1}^H \{W_s(\pi(A))\}$; высота укладки $H = T(\pi(A)) = g_{N+1}^t - 1$, определяющая общее время исполнения всех программных регионов укладки, где g_{N+1}^t

— конечная фиктивная вершина графа. Предложен способ определения наиболее раннего $E(g_i)$ и наиболее позднего $L(g_i, H)$ из возможных сроков выполнения $\forall g_i$ вершины графа, для заданной укладки, на основе расчета длин критических путей на графе. Осуществлена постановка задачи получения оптимальной по времени исполнения укладки программы, как задача минимизации целевой функции: $T(\pi(A)) \rightarrow \min$ на системе ограничений, заданных структурой связей графа и максимальным количеством доступных процессорных элементов p ВС:

$$\left\{ \begin{array}{l} g_i^t < g_j^t - \tau(g_i), \quad \exists (g_i, g_j) \in U, i = \overline{1, N}, j = \overline{1, N}; \\ g_k^t + \tau(g_k), \quad \exists (g_k, g_i) \in U, i = \overline{1, N}, k = \overline{1, N}; \\ g_i^t \geq E(g_i), \quad i = \overline{1, N}; \\ g_i^p \neq g_l^p, \quad \exists t_s \mid g_i \in G(t_s) \wedge g_l \in G(t_s), i = \overline{1, N}, l = \overline{1, N}; \\ W(\pi(A)) \leq p. \end{array} \right.$$

Так как решение данной задачи точными методами математического программирования является весьма трудоемким, в работе предложен эвристический алгоритм, основанный на фиксации начальной укладки, в качестве которой выбирается укладка максимальной степени параллельности. Последующие преобразования ориентированы на приведение ширины укладки к ширине, ограниченной числом процессорных элементов ВС, путём перераспределения вершин p -сечений укладки подлежащих сокращению. Предложенный алгоритм позволяет снизить трудоемкость задачи распараллеливания в сравнении с известными решениями до величины $O(N^2)$.

Так как методы статического синтеза параллельных программ невозможно использовать для программных регионов, имеющих зависимости вида δ_{out-in}^∞ , δ_{in-out}^∞ , $\delta_{out-out}^\infty$, в работе предложено использовать динамический синтез, основанный на методе спекулятивной многопоточности. Чтобы организовать многопоточное выполнение подобных регионов, называемых спекулятивными $\forall g_i \in G^{SP} \mid G^{SP} \subseteq G, i = \overline{1, K}$, формируется динамическая последовательность стадий их исполнения, называемых эпохами $\forall e_j \in EP_i \subseteq EP \mid j = \overline{1, N}, N = |EP_i|, e_j \in \rho_{EP}(g_i), \rho_{EP} : G^{SP} \rightarrow EP$. Например, для циклического региона эпохами являются отдельные итерации цикла. Каждая эпоха снабжается локальным буфером памяти для сохранения критических к возможным нарушениям зависимости данных. Метод предписывает параллельное многопоточное выполнение эпох в расчёте на то, что информационные зависимости на стадии исполнения не проявятся, и будет получен выигрыш в производительности. В случае неудачи результаты вычислений региона должны быть аннулированы, и он будет выполнен повторно, что повлечёт накладные расходы. В работе предложена формализация применения этого метода, сформулированы и доказаны следующие утверждения.

Лемма 3.1. При одновременном параллельном спекулятивном исполнении p эпох $e_j, e_{j+1}, \dots, e_{j+p-1}$, на вычислительной системе состоящей из p процессорных элементов, гарантируется корректность выполнения $\forall e_j$ и отсутствие её повторного перезапуска, где $j, j+1, \dots, j+p-1$ — номера эпох, $e_j, e_{j+1}, \dots, e_{j+p-1} \in EP \mid j = \overline{1, (N-p+1)}, N = |EP|$.

Лемма 3.2. При одновременном параллельном спекулятивном исполнении p эпох $e_j, e_{j+1}, \dots, e_{j+p-1} \in EP \mid j = \overline{1, (N-p+1)}, N = |EP|$ на вычислительной системе состоящей из p процессорных элементов, корректность выполнения $\forall e_{j+k} \mid k = \overline{1, (p-1)}$ гарантируется, если подтверждена корректность $\forall e_{j+k-1}$ эпохи и проверки корректности результатов между каждой парой эпох $\forall (e_{j+k}, e_{j+m}) \mid m = \overline{0, (k-1)}$ дали положительный результат.

Лемма 3.3. Если при одновременном параллельном спекулятивном исполнении p эпох $e_j, e_{j+1}, \dots, e_{j+p-1} \in EP \mid j = \overline{1, (N-p+1)}, N = |EP|$ на вычислительной системе состоящей из p процессорных элементов $\exists e_{j+k} \mid k = \overline{1, (p-1)}$, для которой было зафиксировано нарушение корректности выполнения, то после перезапуска и повторного исполнения e_{j+k} её корректность будет гарантирована.

В работе проведён анализ метода спекулятивной многопоточности, введен ряд параметров, выявлены зависимости между ними и осуществлён расчет оценки выигрыша от использования многопоточной спекуляции для циклических регионов программы:

$$n_{spd}(g_i) = V_{seq}(g_i) / V_{par}(g_i) = \frac{p}{1 + 1/2 \cdot n_{It/E} \cdot Pr_{\delta-} \cdot (p-1) + V_{const} / n_v \cdot n_{It/E}},$$

где $V_{seq}(g_i)$ — объём вычислений циклического региона при последовательном исполнении; $V_{par}(g_i)$ — удельный объём вычислений, приходящийся на один процессорный элемент при параллельном исполнении циклического региона; V_{const} — объём дополнительных вычислений на организацию спекуляции; p — количество процессорных элементов ВС; $Pr_{\delta-}$ — вероятность нарушения зависимости для одной итерации циклического региона; $n_{It/E}$ — число смежных итераций циклического региона g_i , последовательно размещённых внутри каждой эпохи; n_v — число операторов в теле цикла региона g_i .

Предложено использовать $n_{spd}(g_i)$ в качестве критерия оптимальности при формировании спекулятивных регионов и эпох. Разработан алгоритм выделения спекулятивных регионов и эпох на основе МППК, формирующий множе-

ство спекулятивных регионов G^{SP} за один просмотр списка регионов параллельного промежуточного кода, с трудоёмкость порядка $O(N)$, где $N = |G^{SP}|$.

В четвёртой главе представлены результаты экспериментального исследования алгоритмов, предложенных во второй и третьей главах, а также определена область практического применения результатов работы.

Для осуществления исследования разработанных алгоритмов был проведён ряд численных экспериментов, в рамках которых были написаны программы с реализацией алгоритмов на языке программирования Си++. В качестве среды разработки программ использовалось кроссплатформенное программное обеспечение QT 4.6 с компилятором GCC 4.4.0. Результаты численных экспериментов представлены на рис.1 – рис.4.

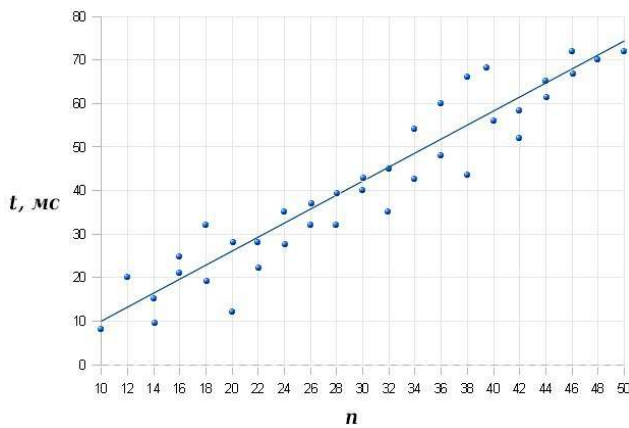


Рис.1. Зависимость времени выполнения программы реализации алгоритма исключения оператора continue из МППК от количества операторов входной последовательности

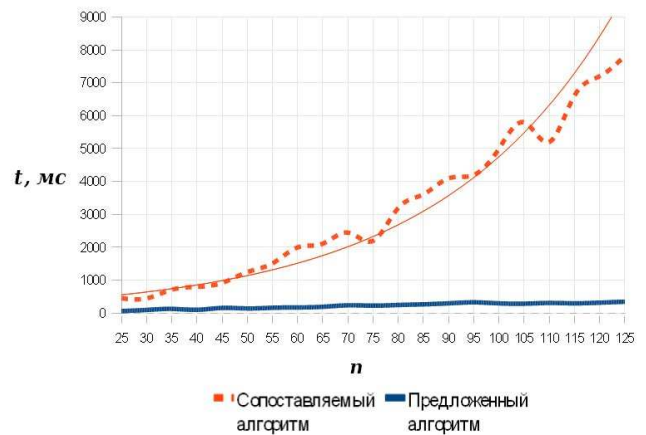


Рис.2. Сравнение времени выполнения программы реализации алгоритма выделения регионов и сопоставляемого с ним от количества операторов входной последовательности

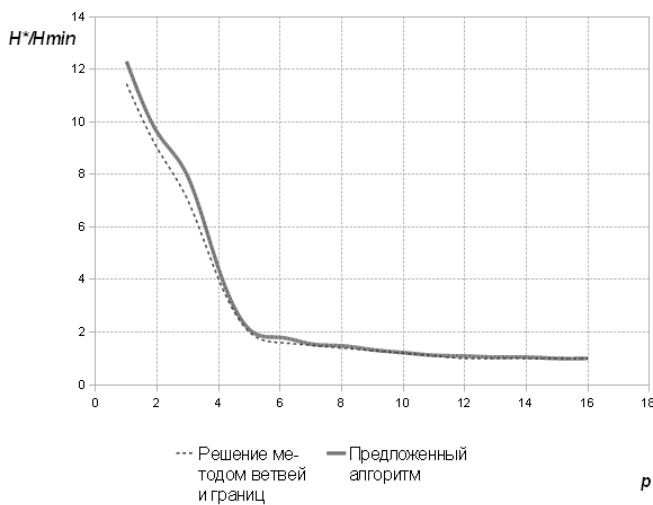


Рис.3. Результаты экспериментальной оценки близости к оптимальным решений эвристического алгоритма статического синтеза параллельной программы при разных значениях p

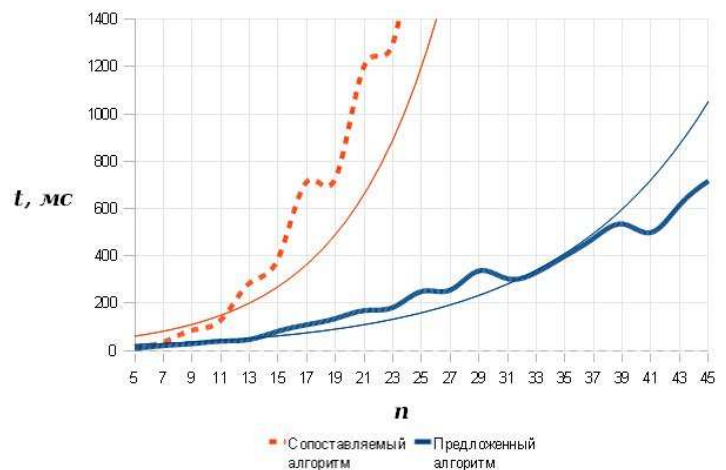


Рис.4. Сравнение времени выполнения эвристического алгоритма статического синтеза параллельной программы и сопоставляемого с ним от числа вершин макрографа программы, на основе их экспериментальной реализации

Другая часть исследований была направлена на определение области практического применения результатов работы. Итогом явилась разработка программных средств поддержки этапа трансляции исходных текстов последовательных программ на языках *Си* и *Паскаль* в МППК, структура которых представлена на рис.6.

Для организации исполнения МППК были разработаны программные модули, представляющие из себя прототип среды исполнения параллельной виртуальной машины, архитектура которой представлена на рис.5.

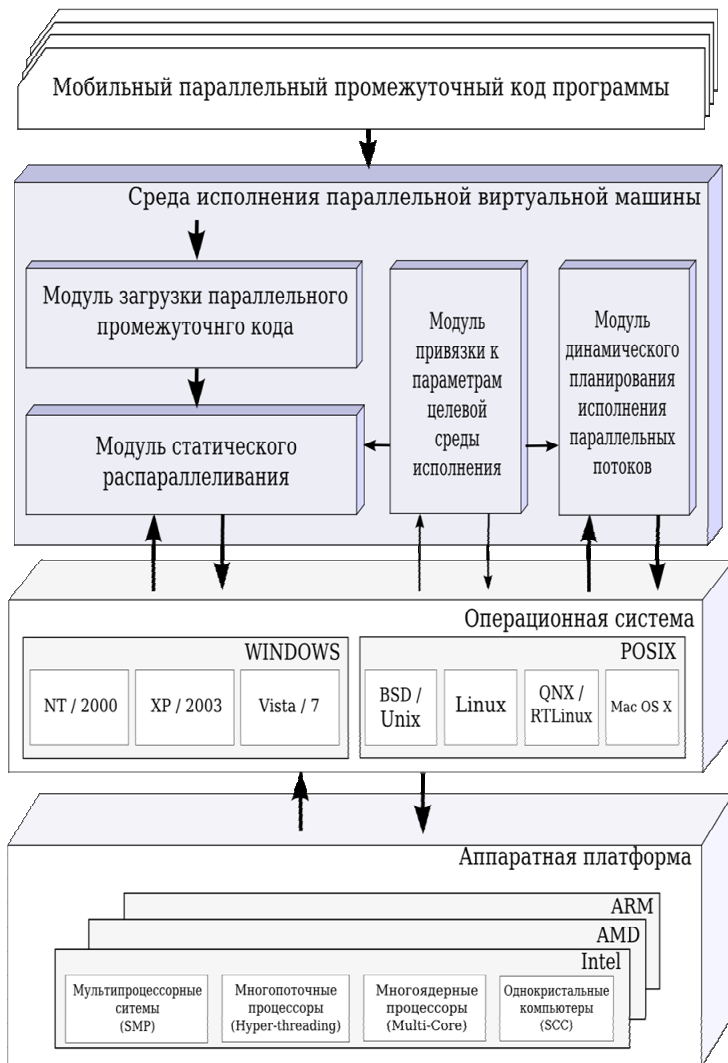


Рис.5. Архитектура параллельной виртуальной машины

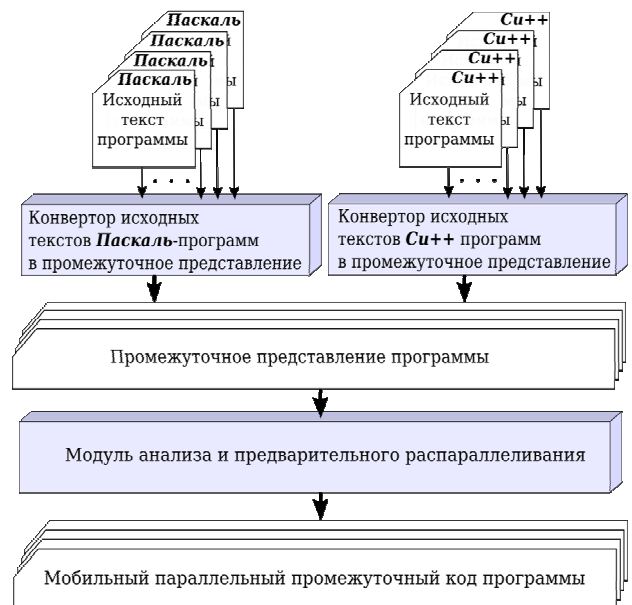


Рис.6. Программные средства преобразования исходных текстов программ в МППК

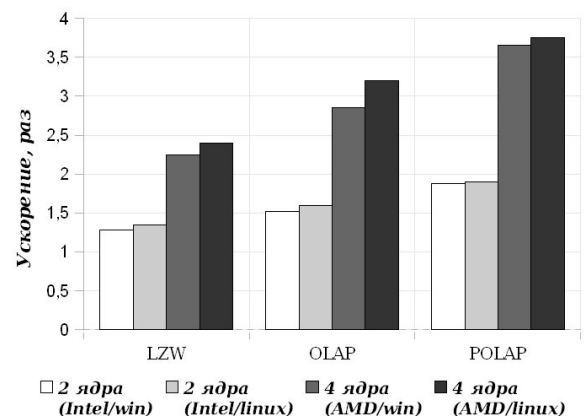


Рис.7. Результаты ускорения, полученного при многопоточном спекулятивном выполнении МППК на различных платформах

С помощью разработанных программных модулей было организовано многопоточное исполнение МППК программы архивации файлов (LZW) и программы реализации алгоритмов оперативного анализа, основанных на кратно-масштабном представлении данных (OLAP), в среде ОС *Window XP* и *Mandriva Linux 2010.1* на ВС с двухъядерным процессором *Intel Core 2 Duo E7200* и 4-х ядерным процессором *AMD Phenom X4 9650*. Результаты сравнения произво-

длительности их выполнения с работой параллельной версии программы реализации алгоритмов оперативного анализа (POLAP), созданной вручную, представлены на рис.7.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Главным итогом диссертационной работы является разработка моделей и алгоритмов, позволяющих организовать высокопроизводительное исполнение последовательных программ в среде современных многоядерных процессоров с учётом обеспечения максимальной мобильности вычислений.

В ходе выполнения работы были получены следующие результаты.

1. Предложена формальная модель мобильного параллельного промежуточного кода, основанная на классе алгебраических моделей типов данных и операций языков программирования, обеспечивающая мобильность параллельного представления последовательных программ, заданных на различных языках высокого уровня, отличающаяся использованием нового базиса управляющих операций.

2. Разработаны алгоритмы исключения операций выхода за пределы ветви управляющей конструкции, основанные на предложенной модели мобильного параллельного промежуточного кода, обеспечивающие корректность представления исходных текстов программ. Разработан алгоритм выделения программных регионов, основанный на предложенной модели мобильного параллельного промежуточного кода, обеспечивающий предварительное крупноблочное распараллеливание программы, отличающийся меньшей трудоёмкостью в сравнении с аналогами.

3. Предложена формальная модель подсистемы управления процессами ОС, основанная на предложенной модели мобильного параллельного промежуточного кода и модели автоматной сети, обеспечивающая представление вычислительного процесса на уровне операционной системы среды многоядерных процессоров.

4. Разработан эвристический алгоритм статического синтеза параллельной программы для мультипроцессорной вычислительной системы с общей памятью, основанный на методе укладки графов, обеспечивающий синтез параллельной программы для ВС с заданным числом процессорных ядер, отличающийся меньшей трудоёмкостью по сравнению с аналогами.

5. Предложена формализация метода спекулятивной многопоточности, используемого для динамического планирования параллельных вычислений, основанная на предложенной модели мобильного параллельного промежуточного кода, обеспечивающая оценку перспективности использования метода спекулятивной многопоточности для циклических участков программы.

6. Разработан алгоритм выделения спекулятивных регионов, основанный на предложенной модели мобильного параллельного промежуточного кода и предложенной формализации метода спекулятивной многопоточности, обеспечивающий повышение эффективности использования данного метода.

7. Экспериментально подтверждена эффективность выполнения предложенного мобильного параллельного промежуточного кода на различных аппаратных и программных платформах с многоядерными процессорами.

ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

1. Бакулев А.В. Автоматическое распараллеливание последовательных программ // Новые информационные технологии: межвуз. сб. науч. тр. Рязань: РГРТА, 1997. С.31-37.
2. Бакулев А.В., Телков И.А. Инструментальные средства создания мобильного программного обеспечения // Проблемы передачи и обработки информации в информационно-вычислительных сетях: тез. докл. 7го междунар. науч.-техн. семинара. - М.: НИЦПрИС, 1997. С. 13-15.
3. Бакулев А.В., Телков И.А. Программные средства организации мобильных параллельных вычислений // Новые информационные технологии. Межвуз. сб. научн. трудов. Рязань: РГРТА, 1997. С. 25-31.
4. Бакулев А.В., Телков И.А. Разработка математической модели и инструментальных средств распараллеливания последовательных программ // Новые информационные технологии в радиоэлектронике: тез. докл. 2-ой всерос. науч.-техн. конф. - Рязань: РГРТА, 1997. С.27-29.
5. Бакулев А.В., Телков И.А. Математическая модель вычислительного процесса симметричной мультипроцессорной системы // Новые информационные технологии в радиоэлектронике: тез. докл. 3-ей всерос. науч.-техн. конф. - Рязань: РГРТА, 1998. С.45-46.
6. Бакулев А.В. Настройка мобильного кода на архитектурные особенности исполнительской системы // Новые информационные технологии: межвуз. сб. науч. тр. Рязань: РГРТА, 1998. С.15-20.
7. Бакулев А.В., Телков И.А. Оптимизация вычислений в SMP-системах // Математическое и программное обеспечение вычислительных систем: межвуз. сб. научн. тр. Рязань: РГРТА, 1999. С. 96-100.
8. Бакулев А.В., Телков И.А. Оптимизация параллельных вычислений в среде MICROSOFT WINDOWS NT // Новые информационные технологии в научных исследованиях и в образовании : тез. докл. 4-ой всерос. науч.-техн. конф. - Рязань: РГРТА, 1999. С.81-83.
9. Бакулев А.В., Телков И.А. Оптимизация параллельных вычислений // Проблемы передачи и обработки информации в информационно-вычислительных сетях: тез. докл. 8-го междунар. науч.-техн. семинара. - Рязань, РГРТА, 1999. С.25-26.
10. Бакулев А.В., Телков И.А. Параллельные процессы в Windows NT // Вычислительные машины, комплексы и сети. Межвуз. сб. научн. трудов. Рязань: РГРТА, 1999. С. 112-115.
11. Бакулев А.В., Телков И.А. Программные средства организации мобильных параллельных вычислений // Управление и информационные технологии на транспорте: тез. докл. междунар. науч.-техн. конф. «Транском-99». - СПб: СПГУВК, 1999. С.195-198.
12. Бакулев А.В., Ручкин В.Н., Телков И.А. Вероятностная модель организации параллельных процессов // МГОУ-XXIII-Новые технологии. Научно-технический журнал. - М, МГОУ, 2000. - №5. - С.35-37.
13. Бакулев А.В., Телков И.А. Конвертор исходных текстов программ с языка Паскаль в мобильный параллельный промежуточный код. М.: Роспатент, 2000 (свидетельство о регистрации программы для ЭВМ №2000610207 от 23.03.2000).
14. Бакулев А.В. Модель операционной среды при организации мобильных вычислений // Новые информационные технологии: межвуз. сб. науч. тр. Рязань, РГРТА, 2000. С. 4-6.
15. Бакулев А.В. Модель организации параллельных процессов в условиях стохастичности // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 5-ой всерос. науч.-техн. конф. - Рязань: РГРТА, 2000. - С.40-41.

16. Бакулев А.В., Телков И.А. Оценка случайной задержки времени исполнения взаимодействующих процессов // Проблемы передачи и обработки информации в информационно-вычислительных сетях: тез. докл. 9-ой междунар. науч.-техн. конф. - Рязань, РГРТА, 2000. С.3-5.
17. Бакулев А.В. Автоматная модель подсистемы управления процессами операционной системы // Новые информационные технологии: межвуз. сб. науч. тр. Рязань: РГРТА, 2001. С.133-137.
18. Бакулев А.В. Модель операций параллельного промежуточного кода // Проблемы передачи и обработки информации в сетях и системах телекоммуникаций: тез. докл. 10-ой междунар. научно-техн. конф. - Рязань, РГРТА, 2001. - С.3-5.
19. Бакулев А.В., Телков И.А. Алгебраическая модель операций мобильного промежуточного кода // Новые информационные технологии: межвуз. сб. науч. тр. Рязань: РГРТА, 2002. С.19-22.
20. Бакулев А.В. Представление управляющих конструкций универсальных языков в алгебраической системе управляющих операций // Новые информационные технологии: межвуз. сб. науч. тр. Рязань: РГРТА, 2002. С.14-18.
21. Бакулев А.В. Повышение производительности выполнения последовательных программ в среде современных многоядерных процессоров // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 11-ой всерос. науч.-техн. конф.- Рязань: РГРТУ, 2006. – С.91-92.
22. Бакулев А.В., Бакулева М.А. Применение вейвлет-преобразования для анализа данных хранилища // Вестник РГРТУ. Научно-технический журнал. Выпуск 21. Рязань: РГРТУ, 2007. С. 57-60.
23. Бакулев А.В., Бакулева М.А., Телков И.А. Алгоритм автоматизации проектирования хранилищ данных // Вестник РГРТУ. Научно-технический журнал. Выпуск 23. Рязань: РГРТУ, 2008. С. 90-93.
24. Бакулев А.В. Использование спекулятивных многопоточных вычислений в среде многоядерных архитектур // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 13-ой всерос. науч.-техн. конф.. - Рязань: РГРТУ, 2008. 140-141.
25. Бакулев А.В., Бакулева М.А. Параллельная реализация алгоритмов оперативного анализа, основанных на кратномасштабном представлении данных хранилища // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 13-ой всерос. науч.-техн. конф. – Рязань: РГРТУ, 2008. С.139-140.
26. Бакулев А.В., Бакулева М.А. Алгоритм исключения операторов выхода за пределы ветви управляющей конструкции в произвольной точке // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 14-ой всерос. науч.-техн. конф. - Рязань: РГРТУ, 2009. – С.191-192.
27. Бакулев А.В. Алгоритм приведения промежуточного кода последовательной программы в соответствие с правилом однократного присваивания // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 14-ой всерос. науч.-техн. конф. - Рязань: РГРТУ, 2009. С. 190-191.
28. Бакулев А.В. Алгоритм синтеза параллельной реализации последовательной программы для вычислительных систем, построенных на базе многоядерных процессоров // Вестник РГРТУ. Научно-технический журнал. Выпуск 30. Рязань: РГРТУ, 2009. С. 43-49.
29. Бакулев А.В. Алгоритм разбиения управляющего графа последовательной программы на слабо связанные регионы // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 15-ой всерос. науч.-техн. конф. - Рязань: РГРТУ, 2010. С. 191-192.
30. Бакулев А.В. Выявление информационных зависимостей между операторами последовательной программы // Новые информационные технологии в научных исследованиях и в образовании: тез. докл. 15-ой всерос. науч.-техн. конф. - Рязань: РГРТУ, 2010. С. 191-192.

Бакулев Александр Валериевич

**МОДЕЛИ И АЛГОРИТМЫ ОРГАНИЗАЦИИ
МОБИЛЬНЫХ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ
В СРЕДЕ МНОГОЯДЕРНЫХ ПРОЦЕССОРОВ**

Автореферат

диссертации на соискание ученой степени
кандидата технических наук

Бумага офисная. Печать цифровая. Усл. печ. л. 1,0.
Тираж 100 экз.

Рязанский государственный радиотехнический университет.
390005, г.Рязань, ул.Гагарина, д.59/1.