

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

УДК 681.3.06:681.324

СОЛОМЕННИК
Михаил Васильевич

МОДЕЛЬ, МЕТОДЫ И СРЕДСТВА
ЭКСТРАКЦИИ, ОПТИМИЗАЦИИ И РЕАЛИЗАЦИИ СЕТЕВОГО
ПАРАЛЛЕЛИЗМА ИЗ ТРУДНО РАСПАРАЛЛЕЛИВАЕМЫХ
ПОСЛЕДОВАТЕЛЬНЫХ АЛГОРИТМОВ

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

по специальности 05.13.11 – Математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

Минск 2009

Работа выполнена в учреждении образования «Белорусский национальный технический университет»

Научный руководитель **Прихожий Анатолий Алексеевич**, доктор технических наук, профессор, заведующий кафедрой программного обеспечения сетей телекоммуникаций Высшего государственного колледжа связи

Официальные оппоненты: **Петровский Александр Александрович**, доктор технических наук, профессор, заведующий кафедрой электронных вычислительных средств, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Дудкин Александр Арсентьевич, кандидат технических наук, ведущий научный сотрудник лаборатории «Идентификации систем», Объединенный институт проблем информатики НАН Беларуси

Оппонирующая организация: Белорусский государственный университет

Защита состоится 26 ноября 2009 г. в 14:00 на заседании совета по защите диссертаций Д 02.15.06 при учреждении образования «Белорусский государственный университет информатики и радиоэлектроники» по адресу: 220013, г. Минск, ул. П. Бровки 6, корп. 1, ауд. 232-1, e-mail: dissovet@bsuir.by, тел. (017) 293-89-89.

С диссертацией можно ознакомиться в библиотеке учреждения образования «Белорусский государственный университет информатики и радиоэлектроники»

КРАТКОЕ ВВЕДЕНИЕ

Проблема параллелизма находится в поле пристального внимания исследователей на протяжении последних десятилетий. В ее рамках сформулирован и решен ряд теоретических и практических задач. Переход от реализации вычислительных алгоритмов и процессов на однопроцессорном компьютере к их реализации на многопроцессорной системе, работающей параллельно, привел к необходимости разработки новых подходов к построению моделей параллелизма и отображению существующих или вновь создаваемых алгоритмов в их адекватные параллельные сетевые версии. На практике важен не только сам переход, но и то, насколько эффективно он выполнен.

Оптимизация в ходе распараллеливания вычислений - важнейшая задача, решаемая в процессе разработки и использования высокопроизводительной информационно-вычислительной системы. Результаты ее решения определяющим образом влияют на конечные параметры системы. Существуют поведенческие описания, на базе которых организация параллельных процессов является наиболее эффективной. Возможности разрабатываемых методов оптимизации определяются размером пространства возможных распараллеливаний алгоритма, которое, в свою очередь, зависит от возможностей используемой модели параллельной системы. Ограниченность модели влечет ограниченность методов оптимизации. Проблема оптимизационного синтеза и реализации сетевых асинхронных алгоритмов на многопроцессорной системе является сложной комбинаторной недостаточно разработанной к настоящему времени проблемой структурного синтеза в многомерном параметрическом пространстве. Таким образом, задача экстракции, синтеза и оптимизации асинхронных параллельных сетевых вычислений по последовательным трудно распараллеливаемым алгоритмическим описаниям является актуальной и практически востребованной.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Связь работы с крупными научными программами

Основные результаты диссертационной работы получены в соответствии с научно-техническими заданиями и планами работы кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» Белорусского национального технического университета, в частности, при выполнении НИР «Исследование проблемно-ориентированного языка VHDL применительно к моделям цифровых систем», № ГР 0002521. Результаты

получены также при выполнении госбюджетной темы «Разработка теории, программных средств и технологий программирования высокопроизводительных телекоммуникационных сетей» на кафедре «Программное обеспечение сетей телекоммуникаций» Высшего государственного колледжа связи.

Цель и задачи исследования

Целью работы явилась разработка новых модели, методов, алгоритмов и программных средств, повышающих эффективность экстракции параллелизма из трудно распараллеливаемых алгоритмов, обеспечивающих оптимизацию реализаций сетевых вычислений.

Для достижения поставленной цели решались следующие задачи:

1. Разработать эффективную модель экстракции параллелизма из трудно распараллеливаемых последовательных алгоритмов.

2. Разработать методы и алгоритмы отображения последовательных вычислений в сетевые алгоритмы.

3. Разработать методы оптимизации сетевых алгоритмов по критериям минимума времени выполнения и стоимости реализации.

4. Создать средства разработки, оптимизации и реализации сетевых алгоритмов.

Объектом исследования явились сетевые алгоритмы и их реализации.

Предметом исследования явились модели, методы и средства распараллеливания, оптимизации и реализации сетевых алгоритмов.

Положения, выносимые на защиту

1. Одноблочная потоковая модель последовательного алгоритма, отличающаяся возможностью экстракции параллелизма из трудно распараллеливаемых последовательных алгоритмов и обеспечивающая повышение предельного уровня распараллеливания их сетевых реализаций.

2. Метод экстракции параллелизма из трудно распараллеливаемых программ, базирующийся на динамических оценках параметров и правилах преобразования программы к одноблочной потоковой модели, отличающийся возможностью редуцирования потока управления и высвобождения потока данных.

3. Метод построения, генерации и редуцирования графа сети асинхронной параллельной модели сетевого алгоритма по впервые предложенной одноблочной потоковой модели с учетом требуемого уровня параллелизма и конвейеризации, имеющихся вычислительных ресурсов.

4. Алгоритмы оптимизации сетевых вычислений, минимизирующие время выполнения или объем используемых вычислительных ресурсов, базирующиеся

на предварительном анализе одноблочной потоковой модели, построении и использовании системы бинарных отношений на множестве элементов модели.

5. Метод трансформации сетевых алгоритмов в VHDL-модель и в параллельный MPI-код. Вычислительные эксперименты показали, что разработанный метод позволяет сократить до двух раз время выполнения сетевого асинхронного алгоритма по сравнению с функционально эквивалентным синхронным последовательно-параллельным алгоритмом.

Личный вклад соискателя

Основные положения и результаты, выносимые на защиту, получены лично автором. Научный руководитель доктор технических наук А. А. Прихожий, соавтор основных публикаций, принимал участие в формулировании целей исследования, постановке задач, определении возможных путей их решения и обсуждении полученных результатов. Вклад Т. Л. Труханович состоит в подготовке примеров экстракции параллелизма в web-приложениях. Вклад Г. Млынека и М. Матавелли заключается в разработке метода динамической оценки вычислительной сложности алгоритма.

Апробация результатов диссертации

Основные положения и результаты диссертационной работы докладывались и обсуждались на следующих международных и республиканских научно-технических конференциях: FDL 2001 - Forum on Design Languages (Леон, Франция, 2001); PARELEC 2002 - Parallel Computing in Electrical Engineering (Варшава, Польша, 2002); Управление в научно-информационной сфере (Минск, Беларусь, 2005), VIII, XI, XII, XIII международных научно-технических конференциях «Современные средства связи» (Минск, Беларусь, 2003, 2006, 2007, 2008); Современная радиоэлектроника: научные исследования и подготовка кадров (Минск, Беларусь, 2006, 2008), Новые информационные технологии в телекоммуникациях и почтовой связи (Минск, Беларусь, 2006, 2008); международной научно-практической конференции «Веб-программирование и Интернет-технологии (WebConf09)» (Минск, Беларусь, 2009).

Опубликованность результатов диссертации

Результаты диссертации опубликованы в 22 печатных работах на 4,74 авторских листах, из них 7 статей в научных журналах, 15 работ в сборниках трудов и материалов международных и республиканских конференций. Объем

публикаций по теме диссертации, соответствующих пункту 18 Положения о присуждении ученых степеней и присвоении ученых званий в Республике Беларусь, составляет 2,45 авторских листа и 2,29 авторских листа составляет объём работ опубликованных в материалах конференций. Результаты диссертации использованы в шести учебно-методических пособиях.

Структура и объём диссертации

Диссертация состоит из введения, общей характеристики работы, пяти глав, заключения, библиографического списка, списка публикаций автора, приложений. В первой главе дается обзор и анализ известных архитектур, моделей, методов, алгоритмов и средств синтеза, оптимизации и реализации параллельных систем. Во второй главе предлагается новая одноблочная потоковая модель программы, исследуются ее свойства, разрабатывается метод экстракции параллелизма из трудно распараллеливаемых алгоритмов. В третьей главе разрабатываются модель сетевых алгоритмов, метод синтеза графов сетевых алгоритмов по одноблочной потоковой модели. В четвертой главе предлагаются алгоритмы оптимизации сетевых вычислений, средства разработки и реализации сетевых алгоритмов. В пятой главе представлены результаты практического применения и экспериментального исследования разработанных модели, методов и средств.

Полный объём диссертации составляет 180 страницы, в том числе 89 страниц основного текста, 99 иллюстраций на 43 страницах, 38 таблиц на 22 страницах, 149 наименований в списке использованных источников на 12 страницах, 14 страниц приложений.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во *введении* определена область и основные направления исследования, обоснована актуальность темы диссертационной работы, дана краткая характеристика направления исследований, а также определена значимость задачи оптимизации многопроцессорных систем.

В *первой главе* выполнен анализ известных моделей, методов и средств построения, синтеза, оптимизации и реализации многопроцессорных систем. Дана классификация архитектур многопроцессорных систем и показано, что известные способы оценки ускорения, достигаемого за счет параллельной реализации алгоритмов, такие как закон Амдала, быстры и эффективны в использовании, но не обеспечивают достаточную точность и не учитывают конкретных параметрических и структурных особенностей алгоритма. Поиск эффективных параллельных реализаций сложных алгоритмических описаний должен учитывать особенности архитектуры и конкретные параметры многопроцессорных систем.

В первой главе дан обзор моделей параллельных вычислений, а также синхронных и асинхронных моделей последовательных алгоритмов, используемых с целью экстракции параллелизма. Известные графовые модели параллельных вычислений, такие, как сеть Петри, граф управления CASCADE, сеть Pt/T-Net (предикат/транзакция) и другие ориентированы на описание параллелизма в сложных управляющих и вычислительных процессах, но не ориентированы на оптимизацию ресурсов параллельной системы. Модель графа задач хотя и освещена широко в литературе и используется как эффективная база для поиска оптимальных отображений задач на процессоры, однако не описывает циклические и ветвящиеся процессы и в силу этого обладает ограниченными возможностями.

Как показывает анализ, известны два глобальных подхода к разработке параллельных систем. Первый состоит в «ручном» создании параллельных описаний с использованием подходящих языков, технологий и сред программирования. Второй базируется на автоматическом распараллеливании и планировании информационно-вычислительных процессов. Каждый из подходов имеет свои преимущества, однако, очевидно, что второй подход, реализуемый посредством распараллеливающих компиляторов, является менее трудоемким и более продуктивным. В его основе лежит экстракция параллелизма из последовательных алгоритмических описаний. Модели и методы экстракции параллелизма хорошо разработаны для так называемых распараллеливаемых программ. Для трудно распараллеливаемых программ методы экстракции параллелизма недостаточно исследованы.

Оптимизация в процессе распараллеливания вычислений и отображения алгоритмов на многопроцессорные системы, формулируемая в литературе как проблема планирования, решающим образом влияет на конечные параметры параллельных реализаций алгоритмов. Задаче оптимизации вычислений на сетях, генерируемых при экстракции параллелизма из последовательного кода, уделено в литературе мало внимания, как следствие, необходима разработка методов оптимизации сетевых алгоритмов по критериям минимума времени выполнения и стоимости реализации.

Вторая глава посвящена разработке одноблочной потоковой модели программы, а также метода экстракции параллелизма из трудно распараллеливаемых последовательных алгоритмов [2–А, 4–А]. Анализ показывает, что одним из ключевых понятий последовательной программы, оказывающим решающее влияние на работу, свойства и возможности моделей и методов экстракции параллелизма, является понятие линейного участка. Преобразование линейных участков и изменение управления их выполнением обеспечивает экстракцию следующих видов параллелизма: внутреннего параллелизма одного или нескольких следующих друг за другом линейных участков; внутреннего параллелизма нескольких следующих друг за другом

информационно зависимых копий одного линейного участка; внешнего параллелизма между линейными участками; внешнего параллелизма между информационно независимыми или информационно зависимыми копиями одного линейного участка. Затруднительной является экстракция параллелизма из циклических/ветвящихся программ, строящихся посредством композиции инструкций *while*, *do-while*, *if*, *switch* и выполняющих обработку данных произвольных типов. Методы экстракции параллелизма из программ этой категории мало разработаны или не изучены вовсе.

В основе предлагаемого в диссертации [1–А, 5–А, 9–А, 13–А] метода экстракции параллелизма из сложного последовательного программного кода лежит представление об итерационном преобразовании программы, улучшающем на каждом шаге ключевые параметры кода, ассоциируемые с его последующими параллельными реализациями. Извлечение из кода параллелизма, улучшающего параметры реализации, является сложным процессом, требующим знания ключевых понятий о статических и динамических свойствах программы. Предлагаемый метод базируется на следующих ключевых положениях: измерении параметров последовательного кода, предсказывающих параметры последующих параллельных реализаций; направленном пошаговом преобразовании посредством правил трансформации последовательного кода с целью улучшения параметров эквивалентного параллельного кода; отображении последовательного кода в эффективные сетевые реализации с использованием одноблочной потоковой модели.

Следующие главные принципы [5–А, 10–А] лежат в основе разрабатываемого и активно используемого в диссертации метода оценивания потенциального параллелизма, скрытого в последовательном алгоритме: профили вычислительной сложности, критического пути и предельного распараллеливания оцениваются динамически на графе исполнения потока данных DFEG в терминах базовых операций языка программирования; потенциал распараллеливания оценивается как вычислительная сложность алгоритма, деленная на длину критического пути.

Правила P1C – P14C предварительного преобразования потока управления исходного алгоритма на языке C представлены в таблице 1. Они позволяют трансформировать, расщеплять и унифицировать такие управляющие структуры, как *if*, *switch*, *while*, *do-while*, *for*, *break*, *continue* и другие [7–А]. Количество путей на графе потока управления алгоритма, определяющее сложность анализа исходного последовательного кода и синтеза параллельного результирующего кода, растет экспоненциально в зависимости от размера графа и определяется количеством базовых блоков (линейных участков) кода алгоритма. Сокращение числа линейных участков приводит к упрощению графа потока управления с точки зрения эффективности распараллеливания.

Таблица 1 – Правила преобразования алгоритма на языке C

№	Фрагмент до преобразования	Фрагмент после преобразования
P1C	<i>while</i> (C) { S }	<i>while</i> (1) { <i>if</i> (C) { S } <i>else break</i> ; }
P2C	<i>do</i> { S } <i>while</i> (C);	<i>while</i> (1) { S <i>if</i> (!C) <i>break</i> ; }
P3C	<i>for</i> (S ₁ ; C; S ₂) { S ₃ }	S ₁ ; <i>for</i> (::) { <i>if</i> (C) { S ₃ S ₂ } <i>else break</i> ; }
P4C	S ₁ <i>if</i> (C) { <i>while</i> (1) { S ₂ } }	V=1; <i>while</i> (1) { <i>if</i> (V) { S ₁ V=0; } <i>if</i> (C) { S ₂ } <i>else break</i> ; }
P5C	S ₁ <i>if</i> (C) { <i>while</i> (1) { S ₂ } }	V=1; <i>while</i> (1) { <i>if</i> (V) { S ₁ } <i>if</i> (!V C) { S ₂ V=0; } <i>else break</i> ; }
P6C	<i>if</i> (C ₁) { <i>break</i> ; } <i>if</i> (C ₂) { <i>break</i> ; }	<i>if</i> (C ₁ C ₂) { <i>break</i> ; }
P7C	<i>if</i> (C) { <i>break</i> ; } S	V=C; <i>if</i> (!V) { S } <i>if</i> (V) { <i>break</i> ; }
P8C	<i>for</i> (S ₁ ; C ₁ ; S ₂) { S ₃ <i>if</i> (C ₂) <i>continue</i> ; S ₄ }	S ₁ <i>for</i> (::) { { <i>if</i> (!C ₁) <i>break</i> ; S ₃ <i>if</i> (!C ₂) { S ₄ }; S ₂ }
P9C	<i>while</i> (1) { S ₁ <i>if</i> (C) <i>continue</i> ; S ₂ }	<i>while</i> (1) { S ₁ <i>if</i> (!C) { S ₂ } }
P10C	<i>if</i> (C) { S ₁ } <i>else</i> { S ₀ }	V ₁ =C; V ₂ =!C; <i>if</i> (V ₁) { S ₁ } <i>if</i> (V ₂) { S ₀ }
P11C	<i>if</i> (V) { Q ₁ ;...Q _k ; }	<i>if</i> (V) { Q ₁ } ... <i>if</i> (V) { Q _k }
P12C	<i>if</i> (V ₁) { <i>if</i> (V ₂) { S } }	V:=V ₁ && V ₂ ; <i>if</i> (V) { S }
P13C	<i>if</i> (V ₁) { V ₂ =E; }	V ₂ = (V ₁ && E) (!V ₁ && V ₂);
P14C	<i>switch</i> (E) { <i>case</i> H ₁ : S ₁ ... <i>case</i> H _n : S _n <i>default</i> : S _{n+1} }	V=E; <i>if</i> (V==H ₁) { S ₁ } <i>else</i> ... <i>if</i> (V==H _n) { S _n } <i>else</i> { S _{n+1} }
M1C	<i>while</i> (1) { S ₁ <i>if</i> (V ₁) <i>break</i> ; S ₂ <i>if</i> (V ₂) <i>break</i> ; S ₃ } S ₄	<i>while</i> (1) { S ₁ <i>if</i> (!V ₁) { S ₂ } V ₃ =V ₁ V ₂ ; <i>if</i> (V ₃) { S ₄ <i>break</i> ; } S ₃ }
M2C	S ₁ <i>while</i> (1) { S ₂ }	V=1; <i>while</i> (1) { <i>if</i> (V) { S ₁ V=0; } S ₂ }
M3C	S ₁ <i>if</i> (C) { <i>while</i> (1) { S ₂ } }	V=1; <i>while</i> (1) { <i>if</i> (V) { S ₁ V=0; } <i>if</i> (C) { S ₂ } <i>else break</i> ; }
M4C	S ₁ <i>if</i> (C ₁) { S ₂ } <i>else</i> { S ₃ <i>while</i> (1) { S ₄ <i>if</i> (C ₂) <i>break</i> ; S ₅ } S ₆ }	V ₂ =1; <i>while</i> (1) { <i>if</i> (V ₂) { S ₁ V ₁ =C ₁ ; <i>if</i> (V ₁) { S ₂ } <i>else</i> { S ₃ } V ₂ =0; } <i>if</i> (V ₁) { S ₄ <i>if</i> (C ₂) { S ₆ <i>break</i> ; } S ₅ } <i>else break</i> ; }
M5C	<i>while</i> (1) { V=1; <i>while</i> (1) { <i>if</i> (V) { S ₁ V=0; } S ₂ <i>if</i> (C) <i>break</i> ; S ₃ } }	V=1; <i>while</i> (1) { <i>if</i> (V) { S ₁ V=0; } S ₂ <i>if</i> (C) V=1; S ₃ }

Модель ОБВМ (One Basic Block Model) включает всего один линейный участок и обеспечивает реальную экстракцию из исходных линейных участков всех видов параллелизма. Схематическое представление каркаса модели ОБВМ на уровне одной функции языка C дано на рисунке 1. Тело единого цикла представляет собой последовательность усеченных операторов *if* с условиями C₁, ..., C_k выполнения операторов Q₁, ..., Q_k. Оператором Q_i может быть оператор присваивания с одноместной или двуместной операцией ⊗, вызов функции с фактическими параметрами e₁, ..., e_n, и операторы *break* и *return*. Модель ОБВМ имеет существенные преимущества по сравнению с известными моделями.

<pre> a) <RType> <FName> (<FArgs>) { <FDeclarations> while (1) { if(C₁) { Q₁ } if(C₂) { Q₂ } if(C₃) { Q₃ } ... if(C_k) { Q_k } } } </pre>	<pre> б) v = ⊗ u; в) v = u ⊗ w; г) v ⊗= u; д) v = f(e₁, ..., e_n); е) break; ж) return expr; </pre>
---	--

Рисунок 1 – Одноблочная модель ОБВМ на языке С

Поскольку при одном прохождении единого цикла выполняется несколько линейных участков исходного кода, в то время как в структурированной модели STRUCT только один участок, общее число итераций цикла ОБВМ всегда меньше, чем в структурированной модели. Единый линейный участок, включающий все операторы исходного кода, обеспечивает полный анализ зависимостей между операторами с последующим выявлением пар распараллеливаемых операторов, пар ортогональных взаимоисключающих операторов, параллельных ветвей в алгоритме, критического пути, позволяющего определить производительность распараллеливаемого кода. За счет экстракции внешнего параллелизма линейных участков коэффициент потенциального распараллеливания одноблочной модели увеличивается, выполнение параллельной реализации кода ускоряется, обеспечивается эффективная конвейеризация алгоритма.

В диссертации предлагается метод [7–А, 18–А] пошагового преобразования произвольного последовательного алгоритма к одноблочной потоковой модели, гарантирующий получение кода модели за конечное число шагов для любого исходного алгоритма. Метод базируется на следующих правилах преобразования, приведенных в таблице 1: M1C – слияние операторов *break*; M2C, M3C, M4C – вставка в цикл операторов, расположенных перед или над циклом; M5C – слияние двух соседних гнездовых циклов в один. Корректность трансформации системы произвольных гнездовых циклов к одному циклу и трансформации последовательности произвольных циклов также к одному циклу доказывается в работе методом индукции. Переход к модели ОБВМ иллюстрируется путем преобразования фрагмента кода стандарта RSA. Замечательным свойством модели является то, что вхождение операторов исходных линейных участков в тело результирующего единого цикла является источником дальнейшей экстракции параллелизма при применении других методов, правил и средств преобразования.

В диссертации разработана методика оценки статических и динамических параметров моделей экстракции параллелизма, показано влияние параметров линейных участков моделей на параметры, характеризующие степень распараллеливаемости алгоритмов. Выполнена оценка параметров четырех моделей экстракции параллелизма для фрагмента кода стандартов RSA и MPEG4 и дано сравнение моделей (рисунок 2) по коэффициентам распараллеливания и ускорения. Коэффициент распараллеливания моделей STRUCT, OBVM, TRASF вырос в 1,38, 1,75 и 2,99 раза по сравнению с исходной моделью SOURCE. В структурной модели STRUCT ускорения вычислений не произошло, в моделях OBVM и TRASF достигнуто ускорение в 1,64 и 2,57 раза.

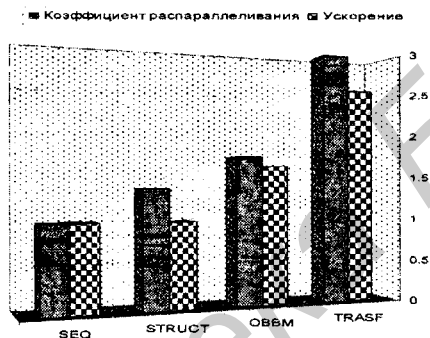


Рисунок 2 – Сравнение моделей экстракции параллелизма

В *третьей главе* предложен метод построения и генерации сетевого алгоритма, в основе которого лежит отображение элементов программного кода одноблочной потоковой модели в маркированные подграфы и увязывания подграфов в корректно работающую сеть. Сетевой алгоритм определяется как $A = (V, O, F, E, M_0)$, где V – множество вершин переменных; O – множество вершин операций; $N = O \cup V$ – множество всех вершин; $F \subseteq N \times N$ – множество дуг, описывающих потоковое отношение; E – множество булевых выражений, ассоциируемых с вершинами; $M_0 \subseteq F$ – начальная разметка дуг. Множество F включает дуги запроса и подтверждения, причём и те и другие могут быть входящими и исходящими для вершин из множества V . Множество дуг запроса, входящих в вершину i , обозначим через *i , множество исходящих дуг – через i^* . Соответствующие множества дуг подтверждения обозначим через *i и i^+ . Дуги могут метиться маркерами. Помеченная дуга является активной, в противном случае она является пассивной. Вершина i может быть помечена булевым выражением $b(i)$. Если $b(i)$ имеет значение true, соответствующая операция выполняется, или соответствующей переменной присваивается значение. Дуга f

также может быть помечена булевым выражением $b(f)$. Если $b(f)$ имеет значение true, движение маркера по дуге разрешено. Две вершины i и j распараллелены в сетевом алгоритме, если в процессе его функционирования существует разметка M дуг такая, что $i^* \cap M \neq \emptyset$ и $j^* \cap M \neq \emptyset$. В противном случае вершины не распараллелены. В зависимости от способа сгорания различаются OR и AND вершины. Граф корректно реализует поток данных одноблочной модели, описывает поток ресурсов, используемых при выполнении алгоритма. Пример сетевого алгоритма дан на рисунке 3. Разработан механизм редуцирования графа, обеспечивающий исключение из него дуг путем вычисления антитранзитивного бинарного отношения, слияние взаимоисключающих ортогональных вершин в агрегированные вершины, другие трансформации сети. Редуцирование графа из рисунка 3 дает граф, изображенный на рисунке 4.

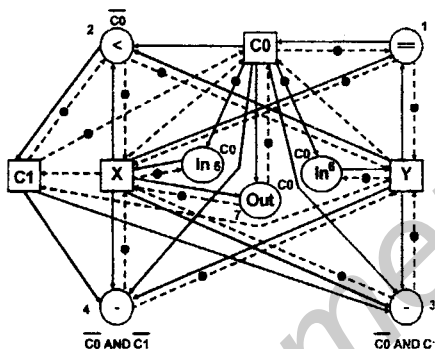


Рисунок 3 – Граф сетевого алгоритма, построенный по модели ОВВМ

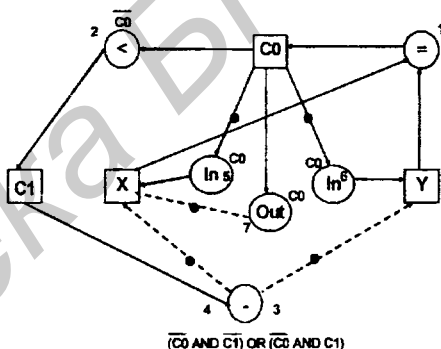


Рисунок 4 – Редуцированный граф сетевого алгоритма

Редуцирование графа сокращает число вершин с 11 до 10, число дуг – с 40 до 12 (в 3,3 раза), число маркированных дуг – с 17 до 5 (в 3,4 раза). Метод трансформации одноблочной потоковой модели и генерации сетевых алгоритмов расширен до синтеза конвейерных сетей, параллельно асинхронно обрабатывающих поток наборов данных.

В четвертой главе показано, что изменение уровня распараллеленности сетевого алгоритма изменяет не только сам граф сети, но также изменяет потоки и объемы ресурсов, потребляемых алгоритмом, которые могут либо увеличиваться, либо уменьшаться. Максимальный уровень распараллеленности оценивается множеством D_m пар параллельных операторов, вычисляемым по одноблочной потоковой модели с использованием бинарных отношений на

множестве управляющих булевых переменных; отношения ортогональности операторов; отношений совместимости и близости операторов; отношения предшествования операторов. Время исполнения сетевого алгоритма оценивается с использованием подмножества $D \subseteq D_M$ с помощью выражения

$$T_D = \max_{q \in Q_D} \left(w(q) \cdot \sum_{i \in q} (n_i \cdot t_{i,p(q)}) \right), \quad (1)$$

где Q_D – множество клик графа G_D ;

$w(q)$ – коэффициент, характеризующий состав операторов в клике $q \in Q_D$ с точки зрения совместимости, близости операций, близости операндов, локализованности операндов в рамках одного процессора и т. д.;

n_i – число исполнений оператора i в процессе выполнения алгоритма;

$p(q)$ – тип процессора, на котором исполняются операторы из клики q ;

$t_{i,p(q)}$ – время исполнения оператора i на процессоре $p(q)$.

Базовым значением величины $w(q)$ является 1. При хорошем группировании операций в клике q величина $w(q) < 1$, при плохом группировании $w(q) > 1$. Стоимость реализации сетевого алгоритма оценивается выражением

$$C_D = \sum_{p \in P} \left(c_p \cdot \max_{q \in Q_D} (z(q) \cdot m_{q,p}) \right), \quad (2)$$

где Q_D – множество клик графа G_D ;

$z(q)$ – коэффициент влияния состава операторов, назначаемых на процессор типа p , на стоимость процессора;

$m_{q,p}$ – число операторов в клике q , назначаемых на процессор типа p .

Базовым значением величины $z(q)$ является единица. Клики графов G_D и G_D оценивают критические пути и критические сечения соответственно на графе сетевого алгоритма.

В четвертой главе сформулированы две оптимизационные задачи в пространстве уровней распараллеленности сетевого алгоритма (рисунок 5). Одна минимизирует время T_D при ограничении C_0 на стоимость, другая минимизирует стоимость C_D при ограничении T_0 на время. Метод минимизации критического пути (сечения) на графе сети, оцениваемого посредством клики максимального веса на графе последовательного (параллельного) выполнения пар операторов, является наиболее эффективным методом оптимизации сетевых алгоритмов. На его базе разработаны статические и динамические градиентные алгоритмы решения задач, выполняющие поиск оптимального

множества D_{opt} путем его пошагового расширения или сокращения (рисунки 5 и 6). Разработано программное обеспечение, написанное на языке C/C++ и ориентированное на использование в распараллеливающих компиляторах и системах высокоуровневого синтеза, позволившее выполнить экспериментальное сравнение алгоритмов оптимизации и определить области их предпочтительного применения.

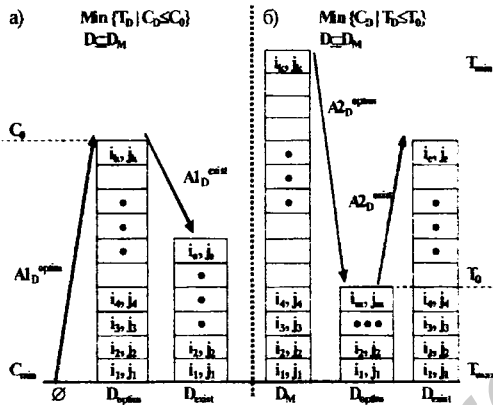


Рисунок 5 – Задачи оптимизации уровня распараллеленности сети

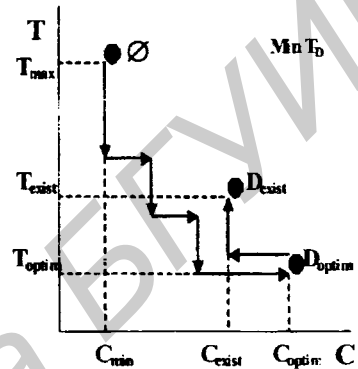


Рисунок 6 – Процесс распараллеливания сети

В четвертой главе разработаны средства визуализированного представления и редактирования сетевых алгоритмов [14–А, 16–А, 18–А, 21–А], построенных посредством базовых конструкций и операторов языка С (рисунки 7). В результате, алгоритмический язык С написания последовательных программ расширен до визуального языка сетевого параллельного программирования. Созданы средства интерпретации и анимации сетевых алгоритмов с целью визуализированной отладки и измерения статических и динамических параметров, включая сложность представления и вычислительную сложность алгоритмов, анимационного исследования динамики поведения сетей, оптимизационных преобразований с изменением степени распараллеленности. Каждый шаг интерпретации сетевого алгоритма разбивается интерпретатором на две ступени: управление и анимация динамики сети, включая движение маркеров и стирание вершин; представление переменных и управление доступом к данным различных типов, включая простые скалярные и сложные составные, реализация всего спектра допустимых языком С операций. Разработаны и реализованы различные стратегии управления динамикой поведения сети: анимация сети путем

перемещения маркеров по дугам и «графического» сгорания вершин; выбор сгораемых вершин «в ширину»; выбор сгораемых вершин «в глубину»; случайный выбор сгораемых вершин; селективное указание и контроль очередной сгораемой вершины. Обеспечивается интерактивное задание количества вершин, сгораемых в процессе интерпретации сети, а также кванта времени, на протяжении которого сеть функционирует, ускорение и замедление функционирования сети. В отличие от известных методов и средств отображения ациклических графов задач на многопроцессорные системы, в диссертации разработан метод генерации и оптимизации кода для компьютерной сети, многоядерного процессора или кластера по графу сетевого алгоритма на базе модели и библиотеки MPI. Предложенный в диссертации метод VHDL-реализации сетевых алгоритмов позволяет отображать граф сети в эквивалентный VHDL-код, обеспечивая исследование, отладку и верификацию сетевых вычислений на ранних стадиях разработки, допуская использование широкого спектра инструментальных средств автоматизированного проектирования и реализации аппаратных и программно-аппаратных параллельных систем.

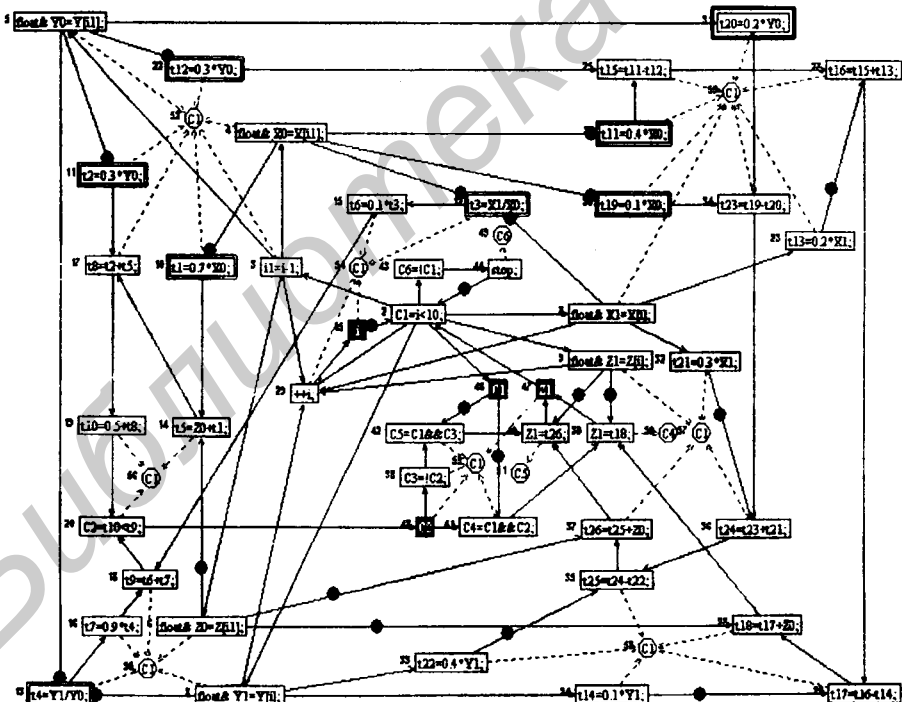


Рисунок 7 – Сетевой алгоритм в среде параллельного программирования

В *пятой главе* рассмотрено применение метода измерений вычислительной сложности, критического пути и потенциала распараллеливания, скрытого в последовательном программном коде одноблочной потоковой модели, для исследования и преобразования программного обеспечения современных аудиокодека ТТА, видеокодека MPEG4 и других сложных алгоритмов. Коэффициент распараллеливания алгоритма MPEG4 составил 1640, алгоритма кодера ТТА – 939, декодера ТТА – 1353. Преобразование кода ТТА увеличило коэффициент распараллеливания кодера до 2051, декодера – до 1696.

На примере аудиокодека ТТА показано, что предложенный метод направленного преобразования программного кода в сетевые модульные алгоритмы (метаалгоритмы) существенно улучшает параметры параллельных реализаций, главным из которых является коэффициент ускорения. Предложенный трансформационный метод построения сетевых алгоритмов обобщен на генерацию конвейерных сетей. Так, переход от сетевого модульного алгоритма кодера ТТА к сетевому двухступенчатому конвейерному метаалгоритму увеличивает коэффициент ускорения с 2,2 до 3,4. Построение матричного сетевого конвейера для решения логического уравнения сокращает время поиска корня во много раз. Выполнены эксперименты с эквивалентными преобразованиями VHDL-кода, показывающие сокращение до 2 раз времени исполнения асинхронных сетевых алгоритмов при варьирующемся времени выполнения операций по сравнению с временем исполнения синхронных алгоритмов при детерминированном времени выполнения операций. Результаты диссертационной работы практически использованы и внедрены (акты о внедрении прилагаются) в ООО «Речевые технологии» при разработке программного обеспечения системы синтеза речи. Результаты полученные при разработке модели, методов и средств распараллеливания и оптимизации сетевых вычислений включены в шесть опубликованных учебно-методических пособий и внедрены в учебный процесс Высшего государственного колледжа связи.

ЗАКЛЮЧЕНИЕ

Основные научные результаты диссертации

1. Установлено, что разработанные к настоящему времени модели и методы распараллеливания алгоритмов и программ ориентированы на класс легко распараллеливаемых задач, описываемых посредством многомерных массивов и систем гнездовых циклов. Модели и методы экстракции параллелизма из

трудно распараллеливаемых алгоритмов разработаны недостаточно и требуют дальнейших исследований и практических реализаций [1–А, 4–А, 7–А].

2. Предложена одноблочная потоковая модель последовательного алгоритма. Модель повышает предельный уровень распараллеливания алгоритма в несколько раз [2–А, 4–А – 8–А, 11–А – 13–А].

3. Предложен метод экстракции параллелизма из трудно распараллеливаемых программ, базирующийся на динамических оценках критического пути и коэффициента распараллеливания на графе исполнения потока данных, а также правилах преобразования, редуцирующих поток управления и высвобождающих поток данных [2–А, 4–А – 8–А, 11–А – 13–А].

4. Разработан метод построения асинхронной параллельной графовой модели сетевого алгоритма по одноблочной потоковой модели с учетом требуемого уровня параллелизма и конвейеризации, имеющихся вычислительных ресурсов [1–А, 2–А, 6–А, 8–А, 9–А, 15–А].

5. Предложены градиентные алгоритмы структурно-параметрической оптимизации сетевых вычислений, минимизирующие время выполнения параллельных сетевых алгоритмов или объем используемых вычислительных ресурсов и базирующиеся на предварительном анализе одноблочной потоковой модели, построении и использовании системы бинарных отношений на множестве элементов модели [2–А, 3–А, 6–А, 9–А, 14–А, 15–А, 22–А].

6. Предложен метод трансформации сетевого алгоритма в VHDL-модель и в параллельный MPI-код. Вычислительные эксперименты показали, что разработанный метод позволяет сократить до двух раз время выполнения сетевого асинхронного алгоритма по сравнению с функционально эквивалентным синхронным последовательно-параллельным алгоритмом [2–А, 6–А, 8–А, 10–А, 19–А].

7. Созданы средства разработки, оптимизации и реализации сетевых алгоритмов, включая язык видео С, средства визуализации, интерпретации и динамического измерения параметров графов сетей, средства оптимизации уровня распараллеленности сетевых алгоритмов, использованные для экстракции параллелизма из стандартного программного обеспечения видеокodeка MPEG4, аудиоcodeка TTA, алгоритма RSA и других сложных алгоритмов [5–А – 7–А, 11–А, 12–А, 16–А – 21–А].

Рекомендации по практическому использованию результатов

Разработанные модель, методы и средства ориентированы на практическое применение в системах программирования многопроцессорных систем. Система параллельного программирования с языком видео С, средства визуализации, интерпретации и оптимизации сетевых алгоритмов поддерживают трансформацию

последовательного алгоритма в форму, эффективную для распараллеливания, обеспечивают измерение параметров и альтернативный просмотр вариантов, расширяют круг лиц, выполняющих разработку и оптимизацию параллельных сетевых вычислений. Разработанные средства могут найти практическое применение на предприятиях, учреждениях для эффективного программирования высокопроизводительных параллельных систем. Результаты диссертационной работы внедрены в ООО «Речевые технологии» при разработке программного обеспечения для анализа и обработки выходных данных системы синтеза речи. Научные результаты диссертации полученные при разработке модели, методов и средств распараллеливания и оптимизации сетевых вычислений внедрены в учебный процесс Высшего государственного колледжа связи.

СПИСОК ПУБЛИКАЦИЙ СОИСКАТЕЛЯ

Статьи в научных журналах

1–А. Прихожий, А.А. Распараллеливание и планирование вычислительных и информационных процессов / А.А. Прихожий, М.В. Соломенник // Доклады БГУИР. – Минск, 2003. – № 4(4). – С. 104–114.

2–А. Прихожий, А.А. Генерирование параллельных сетевых планов по последовательным VHDL моделям / А.А. Прихожий, М.В. Соломенник // Известия Белорусской инженерной академии. – 2003. – № 1(15)/3. – С. 144–146.

3–А. Прихожий, А.А. Оптимизация сетевых алгоритмов / А.А. Прихожий, М.В. Соломенник, М. Маттавелли, Д. Млынек // Известия Белорусской инженерной академии. – 2003. – № 1(15)/3. – С. 147–149.

4–А. Прихожий, А.А. Экстракция параллелизма из последовательного алгоритма путем построения одноблочной потоковой модели программы / А.А. Прихожий, М.В. Соломенник // Известия Белорусской инженерной академии. – 2005. – № 1(19)/4. – С. 120–123.

5–А. Прихожий, А.А. Методология преобразования программного кода с целью улучшения параметров параллельных реализаций аудио кодеков / А.А. Прихожий, М.В. Соломенник, Д. Млынек // Электроника и связь. 2006. – Киев, 2006. – № 2(31). – С. 55–63.

6–А. Прихожий, А.А. Методы и средства распараллеливания, оптимизации, интерпретации и реализации сетевых вычислений / А.А. Прихожий, М.В. Соломенник // Информатизация образования. – Минск, 2008. – № 2. – С. 75–90.

7–А. Соломенник, М.В. Экстракция параллелизма из трудно распараллеливаемых алгоритмов / М.В. Соломенник, А.А. Прихожий, Т.Л. Труханович // Вести Института современных знаний. – 2009. – № 3. – С. 22–30.

Материалы конференций

8–A. Prihozhy, A. Generating Concurrent Net Schedules from Sequential VHDL Models / A. Prihozhy, M. Solomennik // Proceedings of the Fourth International Forum on Design Languages (FDL'2001), September 3–7, 2001, France, Lyon. – Gières: ECSI, 2001. – P. 77–81.

9–A. Prihozhy, A. Techniques for Optimization of Net Algorithms / A. Prihozhy, D. Mlynek, M. Solomennik, M. Mattavelli // Proceedings of the International Conference on Parallel computing in Electrical Engineering (PARELEC'2002). Warsaw, Poland, September 22–25, 2002. – Los Alamitos, California: IEEE Computer Society, 2002. – P. 211–216.

10–А. Соломенник, М.В. Технология реализации параллельных сетевых планов средствами MPI / М.В. Соломенник, А.А. Прихожий // Управление в научно-информационной сфере: материалы Респ. науч.-метод. конф., Минск, 16–17 дек. 2004 г. / Бел. нац. тех. ун-т; под ред. М.М. Болбаса [и др.]. – Минск: УП «Технопринт», 2005. – С. 195–197.

11–А. Соломенник, М.В. Метод отображения последовательного кода аудиокомпрессоров в параллельную реализацию / М.В. Соломенник, Н.Ф. Соломенник, А.А. Прихожий // Современная радиоэлектроника: научные исследования, подготовка кадров: сб. материалов (по итогам работы МНПК, Минск, 20–21 апр. 2006 г.): в 3 ч. Ч. 1. / Минский гос. высш. радиотехн. колледж; под общ. ред. Н.А. Цырельчука. – Минск: МГВРК, 2006. – С. 327–330.

12–A. Prihozhy, A. MPEG-4 Video Codec: Critical Path and Parallelization Potential Analyses / A. Prihozhy, D. Mlynek, M. Mattavelli, M. Solomennik // Современные средства связи: материалы XI междунар. науч.-техн. конф., Нарочь, 25–29 сент. 2006 г. / Высш. гос. колледж связи, Бел. гос. ун-т инф. и радиоэлектрон.; редкол.: М.А. Баркун [и др.]. – Минск: Белпринт, 2006. – С. 88–89.

13–A. Prihozhy, A. Modeling and Analysis of Algorithm Parallelization Potential / A. Prihozhy, M. Mattavelli, D. Mlynek, M. Solomennik, T. Trukhanovich // Современные средства связи: материалы XI международной науч.-техн. конф., Нарочь, 25–29 сент. 2006 г. / Высш. гос. колледж связи, Бел. гос. ун-т информатики и радиоэлектроники; редкол.: М.А. Баркун [и др.]. – Минск: Белпринт, 2006. – С. 90–91.

14–А. Соломенник, М.В. Построение и оптимизация планов сетевых вычислений / М.В. Соломенник, Н.Ф. Соломенник // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы науч.-техн. конф. студ., асп. и молодых специалистов, Минск, 6 дек. 2006 г. / Высш. гос. колледж связи; редкол.: М.А. Баркун [и др.]. – Минск: Бестпринт, 2006. – С. 23.

15–A. Prihozhy, A. Models, Techniques and Tools for Design and Optimization of Multiprocessor Systems / A. Prihozhy, M. Solomennik // Современные средства связи: материалы XII международной науч.-техн. конф., Минск, 24–28 сент.

2007 г. / Высш. гос. колледж связи, Бел. гос. ун-т инф. и радиоэлектрон.; редкол.: М.А. Баркун [и др.]. – Минск: ВГКС, 2007. – С. 163–164.

16–А. Прихожий, А.А. Визуализированная разработка и интерпретация сетевых конвейерных алгоритмов / А.А. Прихожий, М.В. Соломенник, Н.Ф. Соломенник, Т.Л. Труханович // Современная радиоэлектроника: научные исследования и подготовка кадров: сб. материалов (по итогам работы МНПК, Минск, 23–24 апр. 2008 г.): в 3 ч. – Минск: МГВРК, 2008. – Ч. 2. – С. 62–64.

17–А. Соломенник, М.В. Визуализированное распараллеливание и реализация сетевых алгоритмов / М.В. Соломенник, А.А. Прихожий // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы науч.-техн. конф. специалистов, асп., студ., посвященной 15-летию ВГКС, Минск, 14–15 мая 2008 г. / Высш. гос. колледж связи; редкол.: М.А. Баркун [и др.]. – Минск, 2008. – С. 17–18.

18–А. Соломенник, М.В. Визуализированная анимационная разработка и отладка сетевых алгоритмов / М.В. Соломенник, А.А. Прихожий // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы науч.-техн. конф. специалистов, асп., студ., посвященной 15-летию ВГКС, Минск, 14–15 мая 2008 г. / Высш. гос. колледж связи; редкол.: М.А. Баркун [и др.]. – Минск, 2008. – С. 19–20.

19–А. Богонос, А.А. Распараллеливание дискретного преобразования Фурье на базе «МРІ» / А.А. Богонос, М.В. Соломенник, А.А. Прихожий // Новые информационные технологии в телекоммуникациях и почтовой связи: материалы науч.-техн. конф. специалистов, асп., студ., посвященной 15-летию ВГКС, Минск, 14–15 мая 2008 г. / Высш. гос. колледж связи; редкол.: М.А. Баркун [и др.]. – Минск, 2008. – С. 21–22.

20–А. Прихожий, А.А. Методы построения сетевых конвейерных метаалгоритмов / А.А. Прихожий, М.В. Соломенник // Современные средства связи: материалы XIII междунар. науч.-техн. конф., Минск, 7–9 окт. 2008 г. / Высш. гос. колледж связи; редкол.: М.А. Баркун [и др.]. – Минск: ВГКС, 2008. – С. 57–58.

21–А. Соломенник, М.В. Инструментальные средства программирования и оптимизации сетевых алгоритмов / М.В. Соломенник, А.А. Прихожий // Сборник материалов международной научно-практической конференции «Веб-программирование и Интернет-технологии (WebConf09)», Беларусь, Минск, 8–10 июня 2009 г.: в 2 ч. – Минск: Институт математики НАН Беларуси, 2009. – Ч. 1. – С. 33–35.

22–А. Соломенник, М.В. Метод оптимизации сетевых алгоритмов, генерируемых по одноблочной потоковой модели / М.В. Соломенник, А.А. Прихожий // Сборник материалов международной научно-практической конференции «Веб-программирование и Интернет-технологии (WebConf09)», Беларусь, Минск, 8–10 июня 2009 г.: в 2 ч. – Минск: Институт математики НАН Беларуси, 2009. – Ч. 1. – С. 35–36.

М.В. Соломенник

Саламенік Міхаіл Васільевіч

МАДЭЛЬ, МЕТАДЫ І СРОДКІ ЭКСТРАКЦЫІ, АПТЫМІЗАЦЫІ І РЭАЛІЗАЦЫІ СЕТКАВАГА ПАРАЛЕЛІЗМУ З ЦЯЖКА РАСПАРАЛЕЛІВАЕМЫХ ПАСЛЯДОЎНЫХ АЛГАРЫТМАЎ

Ключавыя словы: цяжка распаралеліваемыя алгарытмы, аднаблочная патокавая мадэль, экстракцыя паралелізму, сеткавыя алгарытмы, візуалізаваная распрацоўка, аптымізацыя.

Мэта працы: распрацоўка новых мадэлі, метадаў, алгарытмаў і праграмных сродкаў, якія павышаюць эфектыўнасць экстракцыі паралелізму з цяжка распаралеліваемых алгарытмаў і забяспечаюць аптымізацыю рэалізацыі сеткавых вылічэнняў.

Аб'ектам даследавання з'яўляюцца сеткавыя алгарытмы і іх рэалізацыі.

Прадметам даследавання з'яўляюцца мадэлі, метады і сродкі распаралелівання, аптымізацыі і рэалізацыі сеткавых алгарытмаў.

Атрыманая вынікі і іх навізна. З мэтай рэдукавання патока кіравання, вызвалення патока дадзеных, павышэння лімітавага ўзроўню распаралельвання паслядоўных алгарытмаў прапанавана новая аднаблочная патокавая мадэль праграмы, метады экстракцыі паралелізму з цяжка распаралеліваемых алгарытмаў. Для генерацыі сеткавых вылічэнняў, прадстаўленых асінхроннай паралельнай графавай мадэллю сеткавага алгарытму, прапанаваны метады адлюстравання аднаблочнай патокавай мадэлі ў граф сеткі, які ўлічвае патрабаваны ўзровень паралелізму і канвеерызацыі, наяўныя вылічальныя рэсурсы. Для аптымізацыі графа сеткі прапанаваны градыентныя алгарытмы, якія мінімізуюць час выканання ці аб'ём вылічальных рэсурсаў, што базуюцца на пабудове і выкарыстанні сістэмы бінарных адносін на мностве элементаў аднаблочнай мадэлі. З мэтай генерацыі паралельных рэалізацыі і скарачэння часу іх выканання прапанаваны метады адлюстравання сеткавых алгарытмаў у VHDL- і MPI-код. Створаны сродкі распрацоўкі, аптымізацыі і рэалізацыі сеткавых алгарытмаў на базе мовы C, сродкі візуалізацыі, інтэгрэацыі і дынамічнага вымярэння параметраў графаў сетак, скарыстаныя для экстракцыі паралелізму з такіх складаных алгарытмаў, як RSA, MPEG4, TTA і іншых.

Галіна ужывання: аўтаматызацыя распрацоўкі паралельных прыкладанняў для шматпрацэсарных сістэм і іх апаратнай рэалізацыі.

Соломенник Михаил Васильевич

МОДЕЛЬ, МЕТОДЫ И СРЕДСТВА ЭКСТРАКЦИИ, ОПТИМИЗАЦИИ И РЕАЛИЗАЦИИ СЕТЕВОГО ПАРАЛЛЕЛИЗМА ИЗ ТРУДНО РАСПАРАЛЛЕЛИВАЕМЫХ ПОСЛЕДОВАТЕЛЬНЫХ АЛГОРИТМОВ

Ключевые слова: трудно распараллеливаемые алгоритмы, одноблочная потоковая модель, экстракция параллелизма, сетевые алгоритмы, визуализированная разработка, оптимизация.

Цель работы: разработка новых модели, методов, алгоритмов и программных средств, повышающих эффективность экстракции параллелизма из трудно распараллеливаемых алгоритмов, обеспечивающих оптимизацию реализаций сетевых вычислений.

Объектом исследования являются сетевые алгоритмы и их реализации.

Предметом исследования являются модель, методы и средства распараллеливания, оптимизации и реализации сетевых алгоритмов.

Полученные результаты и их новизна. С целью редуцирования потока управления, высвобождения потока данных, повышения предельного уровня распараллеливания последовательных алгоритмов предложена новая одноблочная потоковая модель программы, метод экстракции параллелизма из трудно распараллеливаемых алгоритмов. Для генерации сетевых вычислений, представленных асинхронной параллельной графовой моделью сетевого алгоритма, предложен метод отображения одноблочной потоковой модели в граф сети, учитывающий требуемый уровень параллелизма и конвейеризации, имеющиеся вычислительные ресурсы. Для оптимизации графа сети предложены градиентные алгоритмы, минимизирующие время выполнения или объем вычислительных ресурсов, базирующиеся на построении и использовании системы бинарных отношений на множестве элементов одноблочной модели. С целью генерации параллельных реализаций и сокращения времени их выполнения предложен метод отображения сетевых алгоритмов в VHDL- и MPI-код. Созданы средства разработки, оптимизации и реализации сетевых алгоритмов на базе языка C, средства визуализации, интерпретации и динамического измерения параметров графов сетей, использованные для экстракции параллелизма из таких сложных алгоритмов, как RSA, MPEG4, ТТА и других.

Область применения: автоматизация разработки параллельных приложений для многопроцессорных систем и их аппаратной реализации.

SUMMARY

Solomennik Mikhail

MODEL, TECHNIQUES AND TOOLS FOR EXTRACTION, OPTIMIZATION AND IMPLEMENTATION OF NET PARALLELIZATION FROM DIFFICULT TO PARALLELIZE SEQUENTIAL ALGORITHMS

Keywords: difficult to parallelize algorithms, one basic block model, parallelism extraction, net algorithms, visualised development, optimization.

The purpose of this work is development of new model, methods, algorithms and tools, to increase parallelism extraction efficiency from difficult to parallelize algorithms and to provide optimization of net computing implementation.

The object of the research are net algorithms and their implementations.

The subject of the research are model, techniques and tools for parallelization, optimization and implementation of net algorithms.

The obtained results and their novelty. New one basic block model of sequential algorithm, parallelism extraction technique from difficult to parallelize programs have been suggested to reduce control flow, to let out of data flow, to increase maximum parallelization level of sequential algorithms. The technique of projection of one basic block model in net graph has been proposed for generation of net computation that is represented as asynchronous concurrent graph model of net graph, that consider required parallelization and pipelining level, available computational resources. The gradient algorithms of structural-parametric optimization have been proposed for net graph optimization. These algorithms minimize execution time or amount of computing resources. They based on preliminary analysis of one basic block model, developing and using binary relation system on the set of model elements. The technique of transformation net algorithms in VHDL- or MPI-code has been proposed for the generation of parallel implementation and decrease implementation time. The tools of development, optimization and implementation of C-based net algorithms, visualization tools, tools for interpretation and dynamic net graph parameters measurement have been created. These tools have been used for parallelism extraction from such complex algorithms as RSA, MPEG4, TTA and other.

Field of application: automation of parallel applications for multiprocessing systems development and their hardware implementation.

Научное издание

СОЛОМЕННИК МИХАИЛ ВАСИЛЬЕВИЧ

**МОДЕЛЬ, МЕТОДЫ И СРЕДСТВА
ЭКСТРАКЦИИ, ОПТИМИЗАЦИИ И РЕАЛИЗАЦИИ
СЕТЕВОГО ПАРАЛЛЕЛИЗМА
ИЗ ТРУДНО РАСПАРАЛЛЕЛИВАЕМЫХ
ПОСЛЕДОВАТЕЛЬНЫХ АЛГОРИТМОВ**

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

**Автореферат диссертации на соискание ученой степени
кандидата технических наук**

Подписано в печать 12.10.2009.	Формат 60x84 ¹ / ₁₆ .	Бумага офсетная.
Гарнитура «Таймс».	Печать ризографическая.	Усл. печ. л. 1,63.
Уч.-изд. л. 1,4.	Тираж 60 экз.	Заказ 658.

Издатель и полиграфическое исполнение: Учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники»
ЛИ №02330/0494371 от 16.03.2009. ЛП №02330/0494175 от 03.04.2009.
220013, Минск, П. Бровка, 6.