

На правах рукописи

Палагин Владимир Владимирович

МАТЕМАТИЧЕСКОЕ И ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ РАСПРЕДЕЛЕНИЯ
ДАННЫХ В ПРОБЛЕМНО-ОРИЕНТИРОВАННЫХ ПАРАЛЛЕЛЬНЫХ
ПРОГРАММАХ

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

Автореферат диссертации на соискание ученой степени
кандидата технических наук

Москва 2014

Работа выполнена на кафедре «Персональные компьютеры и сети» (ИТ-4), факультета «Информатики» Федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Московский государственный университет приборостроения и информатики» (МГУПИ)

Научный руководитель: доктор технических наук, профессор
Баканов Валерий Михайлович,
профессор кафедры «Персональные компьютеры и сети» МГУПИ

Официальные оппоненты: доктор технических наук, профессор
Гостев Иван Михайлович,
профессор кафедры «Управления разработкой программного обеспечения» НИУ ВШЭ
кандидат технических наук, доцент
Рязанова Наталья Юрьевна,
доцент кафедры «Программное обеспечение ЭВМ и информационные технологии»
МГТУ им. Н.Э. Баумана

Ведущая организация: ОАО «Научно-исследовательский институт вычислительных комплексов им. М.А. Карцева»

Защита диссертации состоится 18.06.2014 г. в 16:00 часов на заседании диссертационного Д 212.131.05 в МГТУ МИРЭА по адресу 119454 г. Москва, пр. Вернадского, д.78, ауд. Д412.

С диссертацией можно ознакомиться в библиотеке МГТУ МИРЭА.

Отзывы на автореферат в двух экземплярах, заверенные печатью, просим направлять по адресу: 119454, г. Москва, пр. Вернадского, д. 78, МГТУ МИРЭА, Д212.131.05.

Автореферат разослан 10.05.2014 г.

Ученый секретарь диссертационного совета

кандидат технических наук, доцент

Андрианова Е.Г.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследований. Актуальность исследований в области разработки эффективных параллельных программ в первую очередь определяется необходимостью решения национально-значимых задач в соответствии с приоритетными направлениями развития науки, технологий и техники в Российской Федерации и перечнем критических технологий Российской Федерации, утверждёнными указом Президента от 7 июля 2011 г. № 899. Широкий спектр задач моделирования использует алгоритмы, в которых вычисления производятся во вложенных циклах над значениями функции на границах элементов (узлах сетки). Проблемно-ориентированные параллельные программы, реализующие подобные алгоритмы, требуют значительных вычислительных ресурсов, формируя тем самым большую часть нагрузки на суперкомпьютерные системы.

В то же время основным вектором развития таких систем является повышение максимальной производительности, определяющей научный и технический потенциал отрасли на ближайшие годы и зависящей от быстродействия аппаратной базы вычислителя и эффективности программного обеспечения (ПО). Причём потребность в вычислительных ресурсах не успеваешь решаться только за счёт наращивания мощности аппаратной платформы средствами замены элементной базы или масштабирования.

Эффективность программного обеспечения для многопроцессорных вычислительных систем (МВС) определяется приемлемыми временными рамками обчёта целевой задачи и объёмом потребляемых при этом ресурсов аппаратной платформы. Ручная модификация существующих, и разработка новых параллельных программ эффективных по одному из вышеприведённых критериев является трудоёмкой и затратной процедурой. Использование оптимизирующих/распараллеливающих компиляторов/систем не всегда оправдано с точки зрения проблемы «стена памяти» (*memory wall*), заключающейся в отставании скорости подготовки данных при их передаче по коммуникационной сети от скорости обчёта на вычислительных узлах (ВУ) и

являющейся одним из основных препятствий при достижении экзафлопсного барьера в суперкомпьютерных вычислениях.

Здесь необходимо отметить, что по данным в последние годы акцент в рейтинге проекта TOP500 сместился в сторону вычислительных кластеров – HPC (High-Performance Computing), реализующих архитектуру MPP (Massive Parallel Processing) и занимающих более 84% от общего числа суперкомпьютеров мира с неизменным наращиванием преимущества. В этом контексте ещё более ужесточаются требования к программному обеспечению в части касающейся временной эффективности и ресурсоёмкости.

Одним из путей преодоления отмеченных проблемных мест в области разработки эффективных проблемно-ориентированных параллельных программ для вычислителей с архитектурой MPP является выбор рационального распределения обрабатываемых данных в локальной памяти данных на вычислительных узлах и, как следствие, повышения быстродействия целевой аппаратной платформы MVS. В рамках данной работы под термином проблемно-ориентированные параллельные программы понимаются программы научно-технического характера, обрабатывающие массивы большой размерности, основные вычисления которых сосредоточены во вложенных циклах (гнездах циклов).

В разное время вопросами параллельных вычислений занимались такие исследователи как Воеводин В.В., Воеводин Вл.В., Тыртышников Е.Е., Жуматий С.А., Стефанов К.С., Антонов А.С., Гергель В.П., Вальковский В.А., Кутепов В.П., Соколинский Л.Б., Лацис А.О., Букатов А.А., Дацюк В.Н., Жегуло А.И., Корнеев В.Д., Крюков В.А., Андрианов А.Н., Бугеря А.Б., Ефимкин К.Н.

Наиболее близкими к тематике исследования и посвящённые вопросам размещения данных в памяти вычислителя являются работы Штейнберга Б.Я. и Полуяна С.В. Однако, проблематика распределения данных для проблемно-ориентированных параллельных программ, нацеленных на промышленное использование, со сложной структурой исходного кода, остаётся решённой не полностью. Описанные выше причины вынуждают развивать методы минимизации времени выполнения параллельной программы за счёт модификации

распределения данных в рамках блоков исходного кода с набором гнёзд циклов (*cycle nest*).

Объектом исследования являются МВС с распределённой памятью. Под термином МВС с распределённой памятью понимаются такие классы суперкомпьютеров как кластерные НРС и массово-параллельные МРР вычислительные системы.

Предмет исследования определён областью исследования № 3 паспорта специальности 05.13.11 «Модели, методы, алгоритмы и программные инструменты для организации взаимодействия программ и программных систем», а также перечнем задач, решаемых в диссертации.

Целью работы является уменьшение общего времени выполнения проблемно-ориентированных параллельных программ за счёт снижения потерь времени на пересылку данных между вычислительными узлами средствами рационального размещения массива исходных данных в распределенной памяти МВС.

Задачи исследования. Достижение поставленной цели предполагает решения следующих основных задач:

1. анализ современного состояния проблемы разработки и специфики построения исходного кода проблемно-ориентированных параллельных программ, систематизация методов минимизации времени выполнения без изменения заложенных алгоритмов и аппаратной конфигурации МВС;
2. разработка математической модели, отражающей распределение времени выполнения проблемно-ориентированных параллельных программ с учётом объединения гнёзд циклов, имеющих зависимость по данным, в блоки исходного кода;
3. разработка метода лексико-синтаксического анализа исходного кода проблемно-ориентированных параллельных программ для построения абстрактного синтаксического дерева (АСД) и его декомпозиции, с целью поиска гнёзд циклов со связью по данным и объединения их в блоки;

4. разработка системы эвристических правил, позволяющая производить выбор рационального варианта распределения данных для гнёзд циклов в рамках блоков исходного кода;
5. разработка метода перераспределения времени выполнения между гнёздами циклов со связью по данным в рамках блоков исходного кода;
6. реализация и тестирование препроцессора для выполнения модификации исходного кода проблемно-ориентированных параллельных программ.

Методы исследования включают в себя методы теории компиляции программ, анализа исходного кода, теории графов, теории алгоритмов, технологии разработки программного обеспечения. При реализации программного обеспечения использовались принципы параллельного и функционального программирования.

Научную новизну результатов работы определяют:

1. математическая модель распределения времени выполнения параллельной программы, учитывающая группировку гнёзд циклов с зависимостью по данным в блоки исходного кода;
2. метод лексико-синтаксического анализа параллельной программы для построения абстрактного синтаксического дерева и его декомпозиции, позволяющий производить поиск/обработку гнёзд циклов в блоке;
3. система эвристических правил, позволяющая произвести выбор рационального варианта распределения данных в рамках блоков исходного кода с набором гнёзд циклов;
4. алгоритм и программная реализация балансировки времени выполнения между гнёздами циклов, позволяющие производить модификацию исходного кода проблемно-ориентированных параллельных программ.

Практическая значимость работы состоит в том, что предложенные модели и методы доведены до конкретных алгоритмов, на основе которых была сформирована система эвристических правил и разработано программное обеспечение, посредством которого возможно производить: лексико-синтаксический анализ исходного кода проблемно-ориентированных параллельных программ на метаязыке; разбор функциональных элементов

внутреннего представления препроцессора, сформированного в виде абстрактного синтаксического дерева; модификацию параметра распределения данных в рамках блоков исходного кода.

Практическая значимость работы подтверждается соответствующими актами внедрения.

Достоверность и обоснованность полученных результатов, приведенных в диссертационной работе, обеспечивается:

- использованием надежных методов исследования и результатами, полученными в ходе экспериментов по уменьшению времени выполнения параллельных программ на различных аппаратных платформах МВС;
- апробацией и обсуждением результатов работы на международных и Всероссийских научных конференциях, научными статьями, опубликованными в ведущих научных журналах из перечня ВАК РФ;
- разработкой действующих программных средств, подтвержденных свидетельствами о государственной регистрации.

Внедрение результатов работы. Разработано программное обеспечение «Препроцессор PyNorma», на которое получено свидетельство о государственной регистрации программ для ЭВМ Роспатента №2013612115 от 12.01.2013 г. Данный программный продукт использовался в научно-исследовательских работах в ЗАО «Энергокомплект» (акт о внедрении) и в ОАО «Праймтек» (акт о внедрении). Результаты исследования используются в учебном процессе при подготовке специалистов на кафедре «Персональные компьютеры и сети» МГУПИ в рамках дисциплины «Параллельные вычисления».

Апробация работы. Основные положения и результаты диссертационной работы докладывались на международных и Всероссийских конференциях, и семинарах: международной научно-практической конференции «Современные направления теоретических и прикладных исследований», Одесса, 2008, 2010, 2011, 2012 г.; 2, 3, 4, 5 международной научно-практической интернет-конференции «Актуальные проблемы аппаратно-программного и информационного обеспечения науки, образования, культуры и бизнеса», Москва, 2009, 2010, 2011, 2012 г.; ежегодной Всероссийской научной конференции

«Современные тенденции развития теории и практики управления в системах специального назначения», Москва, 2011, 2012, 2013 г.; 13 Всероссийской научно-технической конференции «Новые информационные технологии», Москва, 2010 г.; научном семинаре «Проблемы современных информационно вычислительных систем» под руководством д.ф.-м.н., проф. В.А. Васенина, МГУ имени М. В. Ломоносова, Москва, 2012 г.; международной заочной научно-практической конференции «Современные тенденции экономики, управления, права, социологии, образования, медицины, физики, математики: Новый взгляд», Санкт-Петербург, 2013 г.;

Публикации. По материалам диссертации опубликованы 21 печатная работа, из них 4 статьи в ведущих научных журналах и изданиях, рекомендованных ВАК РФ, 4 статьи в сборниках научных трудов, зарегистрированных в системе Российского индекса научного цитирования (РИНЦ), 6 публикаций в научных трудах международных и Всероссийских конференций, 7 публикаций в журналах и межвузовских сборниках научных трудов.

Структура и объём работы. Диссертация состоит из введения, 4 глав, заключения, библиографического списка и приложений. В конце работы помещены акты о внедрение результатов работы, и справка о регистрации программ. Общий объём работы – 149 страниц машинописного текста, из них 118 страниц – основное содержание, 19 страниц – библиографический список (168 наименований).

СОДЕРЖАНИЕ РАБОТЫ

Во введении обоснована актуальность темы, сформулированы: цель исследования, научная новизна, практическая ценность работы, основные положения, выносимые на защиту, достоверность и обоснованность научных положений диссертационной работы, апробация результатов исследования.

Первая глава посвящена аналитическому обзору современного состояния проблемы разработки параллельных программ для МВС. Проведён анализ архитектурных особенностей построения аппаратных платформ МВС, основных технологических подходов к разработке параллельных программ, перспектив и

возможностей автоматического распараллеливания. Сформулирован набор требований к параллельным программам. Выбрана стратегия модификации исходного кода проблемно-ориентированных параллельных программ.

Проведён анализ особенностей и специфики современных реализаций архитектур МВС с точки зрения организации памяти и наиболее значимых научных работ в этой области (Флинн М., Фенг Т., Скилликорн Д. и т.д.). Показано, что вектор развития аппаратных платформ для параллельных вычислений промышленного сектора сместился в сторону МВС с распределённой памятью.

Рассмотрены основные технологические подходы к разработке параллельных программ для МВС с общей и распределённой памятью, разобрано использование комбинированных подходов и систем параллельного программирования высокого уровня. Проведён анализ перспектив использования автоматического распараллеливания, его основных преимуществ, ограничений, спектр решаемых задач, существующих программных реализаций.

Сформирован набор требований к параллельным программам, определяющийся промышленным уровнем использования и направленностью современных суперкомпьютеров на обеспечение надежной и экономичной реализации целевого алгоритма при решении прикладных задач, основным из которых остаётся быстродействие.

В результате анализа программно-аппаратного обеспечения современных реализаций параллельных вычислений была сформулирована постановка задачи исследования, а так же ряд ограничений на целевой класс параллельных программ и архитектуру МВС, при использовании которой предложенный подход будет наиболее эффективен. В заключение сделан вывод о необходимости разработки математической модели, методов и алгоритмов уменьшения времени выполнения выбранного класса программ с учётом эффекта «стена памяти».

Вторая глава посвящена разработке математического обеспечения распределения данных в проблемно-ориентированных параллельных программах. Формализована задача исследования, определены основные понятия и термины. Разработана математическая модель распределения времени выполнения для проблемно-ориентированных параллельных программ. Проработан вопрос

использования метаязыковых средств как инструмента модификации исходного кода. Разработан метод лексико-синтаксического анализа исходного кода для проблемно-ориентированных параллельных программ и балансировки времени выполнения между гнёздами циклов на основе системы эвристических правил.

Проработан класс вычислительных задач, реализуемый параллельными программами для научно-технических расчётов, с целью выделения специфики/особенностей построения исходного кода и времени выполнения. Формализовано понятие блока исходного кода, объединяющего в себе одно или несколько гнёзд циклов – структуру вложенных до некоторого уровня циклов. Обоснованием для рассмотрения нескольких гнёзд циклов в рамках блока является зависимость по данным, возникающая в случае доступа по чтению или по записи к одному и тому же массиву переменных. В рамках блока средствами программных механизмов, которые поддерживаются большинством современных систем параллельного программирования (DVM/mpC/НОРМА), распределение данных задаётся единожды и неизменно во время выполнения программы (рис. 1).

| | | |
|--|---|--|
| Распределение данных по ВУ в рамках целевого блока | 1. DISTRIBUTE massive_name format (формат распределения для каждого массива) 2. DISTRIBUTE INDEX i=1..N (формат распределения по индексному направлению) | |
| Начало блока № 1 (beginning of code block) | MASS_X[N,N] | Массив № 1 обрабатываемых данных |
| | DO i=i ₁ , i ₂ , i ₃ (BODY 1,i) DO j=j ₁ , j ₂ , j ₃ (BODY 1,j) DO k=k ₁ , k ₂ , k ₃ BODY 1,k // операции с MASS_*[N][N] END k END j END i | Начало гнезда циклов № 1 глубиной 3 (beginning of cycle nest) |
| | | Конец гнезда циклов №1 |
| | DO i=i ₁ , i ₂ , i ₃ (BODY 2,i) DO j=j ₁ , j ₂ , j ₃ BODY 2,j //операции с MASS_*[N][N] END j END i | Начало гнезда циклов № 2 глубиной 2 (beginning of cycle nest) |
| Конец блока № 1 | | Конец гнезда циклов № 2 |

Рисунок 1 – Структура блока исходного кода с набором гнёзд циклов при формализации задачи поиска рационального распределения данных

Формат распределения зависит от функциональных возможностей используемой системы параллельного программирования или квалификации разработчика в случае использования низкоуровневых библиотек передачи

сообщений. Например, в DVM используется директива *DISTRIBUTE* (вариант 1 на рис. 1), в HOPMA применяется *DISTRIBUTION INDEX* (вариант 2 на рис. 1), при использовании MPI процесс распределения полностью перекладывается на программиста. Заданное распределение действует для некоторой программной области (блока исходного кода) и в её пределах не изменяется.

Распределение данных обозначим как список $D_K = \{d_1, d_2, \dots, d_k, \dots, d_K\}$, где d_k – элемент списка D_K , описывающий заданное распределение, K – общее число вариантов распределения. Блок обозначим, как совокупность гнёзд циклов, и представим в виде списка $F_M = \{f_1, f_2, \dots, f_m, \dots, f_M\}$, где f_m – элемент списка F_M , описывающий цикл m , M – общее число сгруппированных в блоке циклов. Описание гнезда циклов включает ряд параметров, характеризующих тело цикла $BODY_{m,l}$, где m – номер гнезда в блоке, l – глубина вложения.

Согласно принятым обозначениям поиск рационального распределения данных будет соответствовать схеме, приведённой на рисунке 2.



Рисунок 2 – Схема поиска рационального распределения данных в рамках блоков исходного кода на основе системы эвристических правил

Принципиально возможны три подхода к варьированию входных параметров D и F (символами *var* и *const* обозначены действия изменения и постоянства входных параметров соответственно):

1. $D = var, F = const$ – поиск рационального варианта распределения исходных данных при котором времени выполнения блока минимально – подход представленный в данной работе;
2. $D = const, F = var$ – эквивалентные преобразование циклов при неизменном значении распределения в рамках блока – подход характерный для оптимизирующих/распараллеливающих систем (FORGExplorer/V-Ray/OPC);

3. $D = var, F = var$ – одновременная модификация обоих параметров, что является наиболее сложным вариантом оптимизационной задачи.

Тогда общий вид функции при решении задачи уменьшения общего времени выполнения параллельной программы с учётом варьируемых параметров и характеристик аппаратной платформы МВС будет иметь вид: $z = f(D, F, Q_{node}, X_{node}, Y_{net})$, где Q_{node} – число задействованных в процессе обработки данных ВУ, X_{node} – характеристики аппаратного обеспечения вычислительных узлов, Y_{net} – характеристики коммуникационной сети.

Система эвристических правил, позволяющая сделать вывод об рациональности целевого варианта распределения по критерию времени выполнения, представляет собой набор шаблонов выполнения типовых операций, которые наиболее часто используются в проблемно-ориентированных параллельных программах. Типовая операция характеризуется структурой построения гнезда циклов – глубиной вложенности, порядком следования индексов, набором арифметических операций в теле цикла и рядом других параметров (рис 3.).

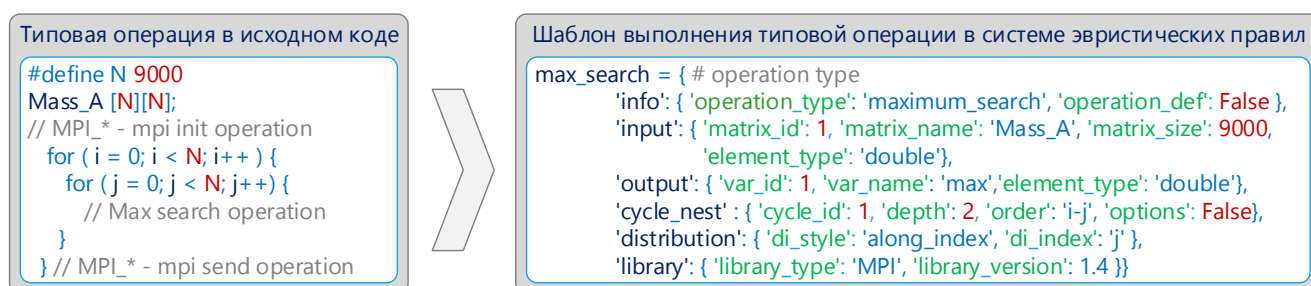


Рисунок 3 – Типовая операция поиска максимального элемента для матрицы в исходном коде и в шаблоне выполнения системы эвристических правил

Шаблон выполнения типовой операции хранит значение рационального распределения данных D для каждой типовой операции, сформированное на основе данных, полученных в ходе практических экспериментов с использованием вычислительных ресурсов кластерной системы МГУПИ и суперкомпьютерного комплекса НИВЦ МГУ им. М.В. Ломоносова. Для каждой тестовой параллельной программы, реализующей одну типовую операцию в рамках блока, опытным путём было получено время выполнения в относительных единицах (для

устранения привязки к конкретной конфигурации аппаратной платформы), зависящее от выбранного параметра распределения данных (рис. 4).

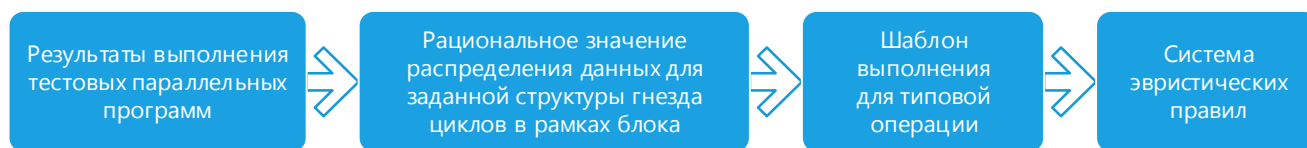


Рисунок 4 – Формирование системы эвристических правил на основе результатов практических экспериментов

В процессе исследования была разработана математическая модель отражающая время выполнения параллельной программы T_{prog} с учётом распределения времени выполнения по блокам T_{block} : $T_{prog} = \sum_{b=1}^B (T_{block,b}) + T_{block}^{ext}$, где $b \in \{1, \dots, B\}$ – число блоков с набором гнезд циклов в программе, T_{block}^{ext} – время выполнения программного кода за пределами блоков. Соответственно, время выполнения блока T_{block} формируется из времени выполнения гнезда циклов с зависимостью по данным T_{nest} и программного кода за пределами циклов T_{nest}^{ext} : $T_{block} = \sum_{m=1}^M (T_{nest,m}) + T_{nest}^{ext}$, где $m \in \{1, \dots, M\}$ – число гнезд циклов с зависимостью по данным в блоке. Исходя из специфики целевого класса программ, для которых основная трудоёмкость сосредоточена в циклах, величины T_{block}^{ext} и T_{nest}^{ext} примем как незначимые с точки зрения времени выполнения.

По причине того, что в каждом теле циклов выполняются как операции индексации при обращении к элементам массивов, так и чисто вычислительные операции, необходимо применять комплексный подход к анализу трудоёмкостей выполнения каждого гнезда циклов.

Для информации о трудоёмкости выполнения гнезда циклов дополним каждый элемент F компонентами, через которые получено время однократного выполнения $t_{m,l}$ тела цикла m глубиной $l \in \{1..l_M\}$: $t_{m,l} = (N_{calc} \cdot T_{calc}) + (N_{mem} \cdot T_{mem})$, где N_{calc} – число арифметических операций, T_{calc} – время выполнения арифметической операции, N_{mem} – число операций индексирования (доступа к данным), T_{mem} – время выполнения операции индексирования. Величины N_{calc} и N_{mem} определяются реализацией целевого алгоритма, в то время

как T_{mem} определяется расположением данных в локальной памяти вычислительного узла или в памяти другого узла.

Полное время выполнения $T_{m,l}$ тела цикла m глубиной l : $T_{m,l} = t_{m,l} \cdot (i_2 - i_1)/i_3$, где i_1, i_2, i_3 – начальное, конечное значение и приращение индексной переменной. В случае неполного выполнения гнезда циклов при использовании операторов условного перехода формализация значительно усложняется. Время выполнения гнезда циклов $T_{nest,m}$ без учёта досрочного выхода из цикла будет определяться: $T_{nest,m} = \sum_{s=1}^{l_m} \left[\prod_{n=1}^{l_m} \left(T_n \cdot (i_{2,n} - i_{1,n}) / i_{3,n} \right) \right]$, где T_n – время выполнения тела цикла на глубине $n(s)$ целевого гнезда циклов.

Для того, что бы обеспечить независимость от аппаратной реализации МВС величины T_{calc} и T_{mem} задаются в относительных единицах времени выполнения. Тогда, определив T_{mem} для каждой операции доступа к данным по индексу при заданном распределении D , получим функцию цели для поставленной задачи $T_{block} = f_B(d_1, d_2, \dots, d_k) \rightarrow min$. Таким образом, уменьшение времени выполнения целевого блока параллельной программы достигается поиском минимума в конечном списке D , хранящем все варианты распределения. Программной реализацией данного алгоритма является препроцессор для метаязыка, позволяющий производить модификацию исходного кода проблемно-ориентированной параллельной программы.

В рамках диссертационного исследования подробно рассмотрены реализации механизмов распределения данных в современных системах параллельного программирования (DVM/мрС/НОРМА). По причине невысокой трудоёмкости лексико-синтаксического анализа по сравнению с классическими языками программирования в качестве инструмента модификации исходного кода целевого класса программ выбран язык НОРМА, разработанный в ИПМ РАН им. М.В. Келдыша. Рассмотрены основные этапы проектирования и разработки, типовая структура исходного кода, функциональные операторы и синтаксические конструкции.

В ходе выполнения работы разработан метод лексико-синтаксического анализа исходного кода параллельной программы на метаязыке. В процессе анализа из входной последовательности выделяются необходимые лексемы (токены) и строится абстрактное синтаксическое дерево (АСД), в котором внутренние вершины сопоставлены с блоками, а листья – с гнёздами циклов. Таким образом, целевая параллельная программа представляется в виде иерархии блоков исходного кода с набором гнёзд циклов, где вершиной является точка входа в программу (рис. 5).

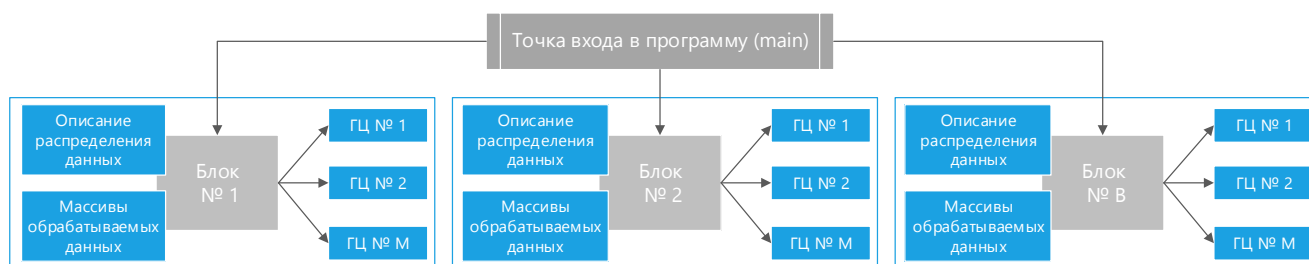


Рисунок 5 – Представление параллельной программы в виде иерархии блоков в процессе лексико-синтаксического анализа

В процессе разбора целевой программы АСД дополняется такой служебной информацией как глубина вложения циклов, порядок следования индексов, которая используется на этапе декомпозиции и балансировки времени выполнения между гнёздами циклов, в ходе которых происходит разбор целевых функциональных элементов.

Инициировать процесс балансировка времени выполнения необходимо в том случае, если в рамках одного блока объединяется набор циклов с зависимостью по данным (рис. 6).

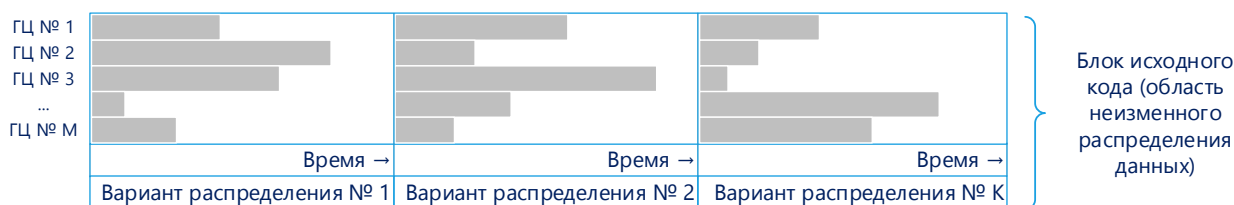


Рисунок 6 – Влияние варианта распределения исходных данных по ВУ на время выполнения ГЦ в блоке

Рациональный вариант распределения по критерию времени выполнения для конкретного гнезда циклов может оказаться нерациональным для других циклов в

рамках целевого блока, что приведёт к эффекту противоположному искомому. Задача может быть упрощена за счёт использования принципа выбора наиболее значимых по трудоёмкости гнёзд циклов. В этом случае определяем вес P_m каждого гнезда m в блоке b согласно: $P_m = T_{nest,m}/T_{block,b}$. Таким образом, становится возможным произвести выбор рационального распределения D только для цикла с максимальным весом $T_{nest,p} = F_P(d_1, d_2, \dots, d_k) \rightarrow \min$, где номер гнезда цикла $m = p$. Адекватность применения веса ГЦ и его времени выполнения определяются правильностью оценки каждого T_{mem} для заданного распределения D , где ключевым моментом является система эвристических правил на основе шаблонов выполнения типовых операций.

Третья глава посвящена разработке препроцессора «PyNorma» для модификации исходного кода проблемно-ориентированных параллельных программ на метаязыке. Приводится описание основного управляющего скрипта программы и специализированных библиотек: ввода/вывода и хранения конфигурации, лексико-синтаксического анализа, работы с абстрактным синтаксическим деревом, балансировки времени выполнения гнёзд циклов, генерации тестов.

Входной последовательность для препроцессора является набор файлов с исходным кодом на метаязыке, который в процессе работы модифицируется и передаётся транслятору НОРМА для генерации результирующей параллельной программы на Си/Фортран с инкапсуляцией вызовов одной из поддерживаемых библиотек передачи сообщений (MPI/GNS/PVM). Затем следует этап компиляции/линковки стандартными средствами разработки и запуск на целевой аппаратной платформе МВС. В зависимости от флагов, с которыми был запущен препроцессор, результатом обработки входного потока будет один модифицированный файл с рациональным распределением данных по вычислительным узлам или набор файлов со всеми возможными вариантами распределения данных в блоке для последующего тестирования или дополнения системы эвристических правил.

Четвёртая глава посвящена тестированию препроцессора «PyNorma» на различных аппаратных платформах МВС. Приведено описание технических характеристик и программного обеспечения выбранных задействованных МВС. Обсуждаются результаты тестирования препроцессора и формирование системы эвристических правил на вычислительном кластере МГУПИ и суперкомпьютере «Ломоносов».

Для получения предварительных результатов и отладки проблемно-ориентированных программ для научно-технических расчётов с набором типовых операций был задействован вычислительный кластер МГУПИ. Полученные в ходе экспериментов данные легли в основу системы эвристических правил препроцессора, которая используется в процессе вычисления рациональных значений распределения данных и ожидаемого коэффициента ускорения для балансировки времени выполнения гнёзд циклов в рамках блока.

По причине того, что реальные параллельные программы для научно-технических расчётов, обрабатывающие данных большого объёма имеют весьма сложную логику и структуру исходного кода, для оценки эффективности препроцессора в реальных условиях были проведены эксперименты с более сложными параллельными программами на аппаратной платформе НИВЦ МГУ им. Ломоносова.

В качестве примера рассмотрим результаты эксперимента по модификации исходного кода параллельной программы, реализующую набор операций над двумерными/одномерными массивами (умножение матриц/векторов, поиск максимального/минимального элемента, транспонирование и т.д.), функционально разнесённым по блокам. Характеристикой качества выбран коэффициент ускорения параллельной программы по времени выполнения принятый, как отношение наихудшего времени выполнения программы к полученному.

На графике группой *A* обозначен пример двух вариантов распределения, иллюстрирующий максимальное ускорение при значениях параметра распределения в блоках $jik-ji-ji/jki-ij-ij$, группой *B* – пример двух вариантов распределения, демонстрирующий относительно незначительное ускорение при значениях распределения $ikj-ij-ij/ijk-ij-ij$, группой *C* – пример двух худших

вариантов распределения без ускорения и принятый за точку отсчёта при значениях распределения $kji-jj-jj/kij-jj-jj$ (рис. 7).

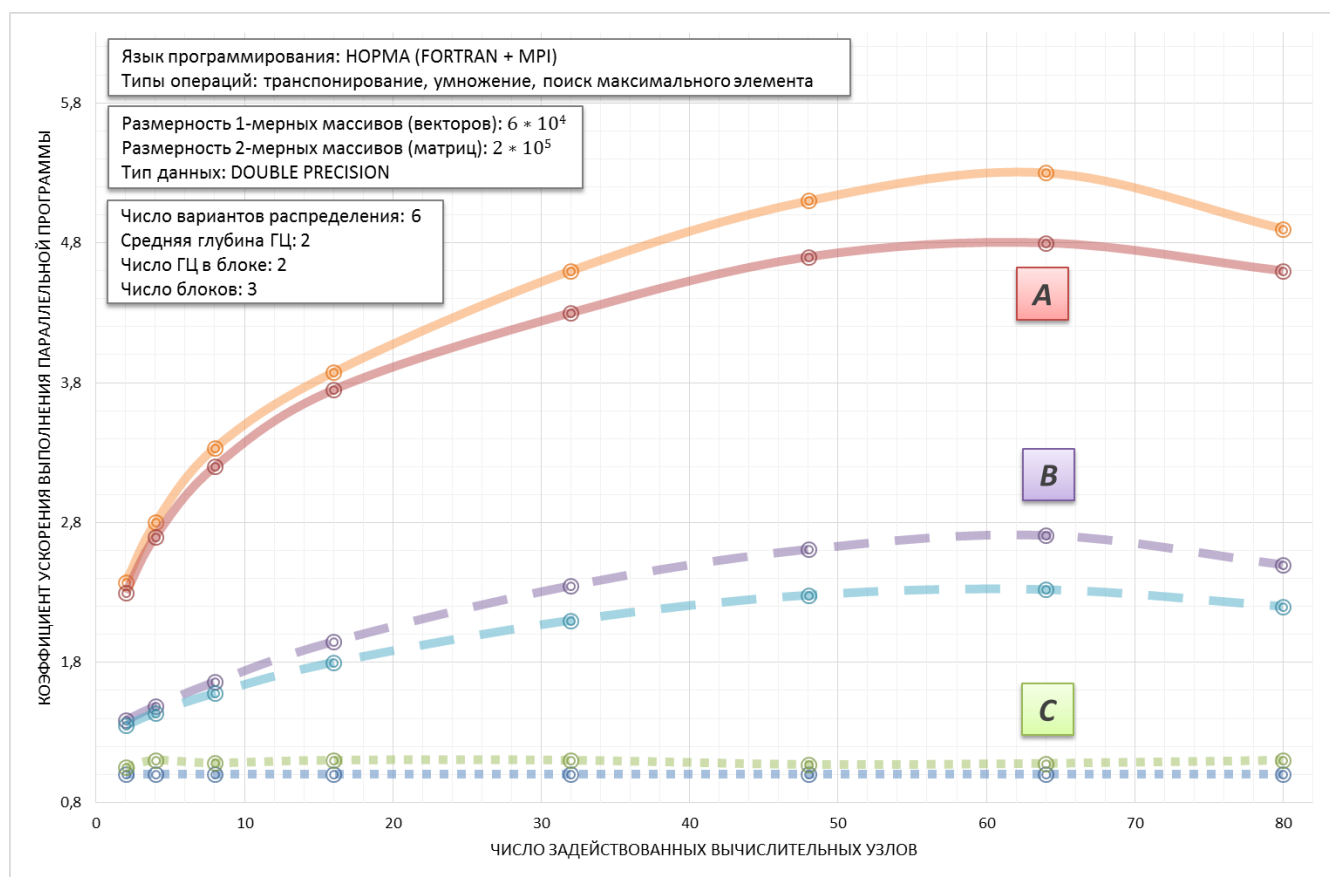


Рисунок 7 – Зависимость коэффициента ускорения выполнения программы от распределения данных на аппаратной платформе суперкомпьютера НИВЦ МГУ им. Ломоносова

В случае целевой программы, которая по структуре исходного кода эквивалента реальным программам для научно-технических расчётов, выполняющихся на МВС промышленного уровня, после препроцессирования исходного кода удалось добиться ускорения в среднем в 3,5-4,9 раз в зависимости от числа задействованных вычислительных узлов.

В Заключение обобщены основные научные и практические результаты, отмечена их научная новизна и практическая значимость. Приведен перечень основных научных результатов, полученных автором лично и выносимых им на защиту.

НА ЗАЩИТУ ВЫНОСЯТСЯ СЛЕДУЮЩИЕ РЕЗУЛЬТАТЫ

1. Математическая модель распределения времени выполнения параллельной программы, с учётом объединения гнёзд циклов, имеющих зависимость по данным, в блоки исходного кода. Модель построена с учётом специфики структуры исходного кода проблемно-ориентированных параллельных программ.
2. Метод лексико-синтаксического анализа исходного кода параллельных программ для построения абстрактного синтаксического дерева и его декомпозиции. Метод позволяет производить поиск гнёзд циклов и группировку их в блоки по принципу наличия связи по данным с целью дальнейшей обработки через систему эвристических правил.
3. Система эвристических правил, позволяющая производить выбор рационального варианта распределения обрабатываемых данных по вычислительным узлам для целевого гнёзда циклов в рамках блока исходного кода параллельной программы. Система эвристических правил состоит из 28 шаблонов выполнения типовых операций, характерных для проблемно-ориентированных параллельных программ промышленного уровня.
4. Метод балансировки времени выполнения между гнёздами циклов со связью по данным в блоках исходного кода, позволяющий определить рациональное, с точки зрения минимизации времени выполнения блока, распределение обрабатываемых данных по вычислительным узлам.
5. Алгоритм и программная реализация специализированного препроцессора метаязыка для выполнения модификации исходного кода проблемно-ориентированных параллельных программ промышленного уровня.
6. Экспериментально обоснованные результаты и рекомендации по модификации исходного кода целевого класса проблемно-ориентированных параллельных программ средствами разработанного препроцессора для метаязыка.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ

В журналах, входящих в перечень изданий, рекомендуемых ВАК.

1. Палагин В.В. Синтаксический анализ исходного кода на языке программирования НОРМА с целью оптимизации распределения данных в параллельных программах для научно-инженерных расчётов. // Современные тенденции развития теории и практики управления в системах специального назначения: Научно-технический сборник ОАО Концерн «Системпром». – М., 2012. – №1(2) – С. 559-566.
2. Палагин В.В. К вопросу об ускорении параллельных программ научно-технических расчётов путём использования архитектурных особенностей многопроцессорных вычислительных систем. // Программная инженерия. – 2013. – №2. – С. 21-25.
3. Палагин В.В. Препроцессирование исходного кода параллельных программ для научно-технических расчётов с целью оптимизации по времени выполнения. // Современные тенденции развития теории и практики управления в системах специального назначения: Научно-технический сборник ОАО Концерн «СистемПром». – М., 2013. – №2(4) – С. 421-429.
4. Палагин В.В., Баканов В.М. Повышение эффективности параллельных программ по времени их выполнения за счёт рационального размещения данных в распределённой памяти вычислителя. // Программная инженерия. – 2014. – №5. – С. 6-13.

В сборниках научных трудов, зарегистрированных в базе РИНЦ.

5. Палагин В.В. Обзор перспективных отечественных технологических направлений разработки параллельных программ для научно-инженерных расчётов. // Современные направления теоретических и прикладных исследований 2011: Сборник научных трудов по материалам международной научно-практической конференции. – Одесса, 2011. – Том 5. Технические науки. – С. 11-14.
6. Палагин В.В. Снижение времени выполнения параллельных программ за счёт рационального использования особенностей архитектуры

многопроцессорных вычислительных систем. // Современные направления теоретических и прикладных исследований 2012: Сборник научных трудов по материалам международной научно-практической конференции. – Одесса, 2012. – Выпуск 1. Том 3. – С. 76-79.

7. Палагин В.В. Использование препроцессора PyNorma с целью оптимизации параллельных программ для научно-технических расчётов по времени выполнения. // Современные тенденции экономики, управления, права, социологии, образования, медицины, физики, математики: сборник научных статей по итогам международной заочной научно-практической конференции. – СПб., 2013. – С. 230-233.
8. Палагин В.В. Разработка системы правил препроцессора с целью оптимизации параллельных программ научно-технического характера по времени выполнения для вычислителей архитектуры MPP. // Вестник МГТУ МИРЭА. – 2014. №1(2) – С. 234-244.

В других изданиях.

9. Палагин В.В. Оптимизация процесса индексирования с целью наилучшего использования кэширования в системе NORMA. // Актуальные проблемы аппаратно-программного и информационного обеспечения науки, образования, культуры и бизнеса: Сборник научных трудов по материалам 2 международной научно-практической конференции. – М., 2009. – С.35-38.
10. Палагин В.В. Современные проблемы оптимизации процесса распараллеливания вычислений при трансляции программ на языке NORMA. // Новые информационные технологии: Сборник трудов XIII Всероссийской научно-технической конференции. – М., 2010. – С. 54-59.
11. Палагин В.В. Актуальные проблемы разработки и оптимизации параллельных программ на языке NORMA. // Современные направления теоретических и прикладных исследований 2010: Сборник научных трудов по материалам международной научно-практической конференции. – Одесса, 2010. – Том 2. Технические науки. – С. 18-22.
12. Палагин В.В., Баканов В.М. Перспективы использования функционального программирования в разработке параллельных программ для

многопроцессорных вычислительных систем. // Программное и информационное обеспечение систем различного назначения на базе персональных ЭВМ: Межвузовский сборник научных трудов – М., 2011. – Выпуск 14. – С. 39-44.

13. Палагин В.В. Проблематика балансировки вычислительной трудоёмкости многоуровневых гнёзд циклов с учётом затрат на передачи данных между вычислительными узлами кластерной системы. // Актуальные проблемы аппаратно-программного и информационного обеспечения, культуры и бизнеса: Сборник научных трудов по материалам 4 международной научно-практической конференции интернет-конференции. – М., 2011. – С. 56-62.
14. Палагин В.В. Специфика отображения параллельной программы на языке программирования НОРМА на архитектуру многопроцессорных вычислительных систем. // Современные тенденции развития теории и практики управления в системах специального назначения: Научная конференция в честь 20-летия ФГУП концерн «Системпром» (Москва, 15 мая 2011 г.) – М., 2011. – С. 222-224.
15. Палагин В.В. Оптимизация по времени выполнения проблемно-ориентированных параллельных программ для научно-инженерных расчётов. // Программное и информационное обеспечение систем различного назначения на базе персональных ЭВМ: Межвузовский сборник научных трудов – М., 2012. – Выпуск 15. – С. 84-89.

Патенты, свидетельства на программы для ЭВМ.

16. Свидетельство о государственной регистрации программы для ЭВМ Роспатента №2013612115 от 12.01.2013.