На правах рукописи

Перепёлкин Владислав Александрович

ЯЗЫК И СИСТЕМА LUNA АВТОМАТИЧЕСКОГО КОНСТРУИРОВАНИЯ ПАРАЛЛЕЛЬНЫХ ПРОГРАММ ЧИСЛЕННОГО МОДЕЛИРОВАНИЯ ДЛЯ МУЛЬТИКОМПЬЮТЕРОВ

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

АВТОРЕФЕРАТ

диссертации на соискание ученой степени  
кандидата технических наук

Новосибирск — 2021

Работа выполнена в Федеральном государственном бюджетном учреждении науки Институт вычислительной математики и математической геофизики Сибирского отделения Российской академии наук

Научный руководитель: Малышкин Виктор Эммануилович

доктор технических наук, профессор

Официальные оппоненты:

Ведущая организация:

Защита состоится «\_\_» \_\_\_\_\_\_\_20\_\_ г. в \_\_:\_\_ на заседании диссертационного совета Д 003.061.02 при Институте вычислительной математики и математической геофизики Сибирского отделения Российской академии наук по адресу: проспект академика Лаврентьева, 6, 630090, г. Новосибирск, Россия.

С диссертацией можно ознакомиться в библиотеке и на сайте ИВМиМГ СО РАН, http://icmmg.nsc.ru.

Автореферат разослан «\_\_»\_\_\_\_\_\_\_20\_\_ г.

Ученый секретарь

диссертационного совета Д 003.061.02,

д.ф.-м.н. Сорокин Сергей Борисович

**Общая характеристика работы**

**Актуальность темы.** Разработка параллельных программ численного моделирования на суперЭВМ требует решения комплекса проблем, связанных с организацией параллельной обработки данных: декомпозиция данных и вычислений, программирование коммуникаций, синхронизация работы процессов и потоков, распределение данных и вычислений по узлам. В более сложных случаях требуется обеспечение динамической балансировки нагрузки на вычислительные узлы, сохранение контрольных точек, обеспечение отказоустойчивости и пр. Для эффективного использования аппаратных ресурсов параллельная программа должна быть переносима, настраиваться на особенности вычислителя: производительности вычислительных узлов, топологию и производительность сети, доступный объём памяти, динамику загрузки узлов. Программа должна быть масштабируемой на большое количество вычислительных узлов и использовать все доступные ресурсы вычислительной системы.

Решение этого круга проблем требует специальной квалификации и значительных трудозатрат, что обуславливает актуальность создания и применения средств автоматического конструирования численных параллельных программ с требуемыми свойствами.

**Объектом исследования** является процесс автоматического конструирования параллельной программы по её высокоуровневой спецификации. **Предмет исследования** — языки и алгоритмы автоматического конструирования и исполнения параллельных программ.

**Степень разработанности темы.** В общей постановке проблема конструирования эффективной параллельной программы по высокоуровневому описанию сложна, поэтому исследуются различные частные и эвристические подходы, обеспечивающие конструирование параллельных программ удовлетворительного качества в частных случаях.

В России по теме работали Ю.И. Янов, В.Е. Котов, В.А. Вальковский, В.Э. Малышкин, Э.Х. Тыугу, А.Л. Фуксман, И.Б. Задыхайло, А.Н. Андрианов, В.А. Крюков, С.М. Абрамов, А.А. Московский и др. За рубежом проблематика разрабатывалась авторами C. Hoare, L. Lamport, L. Kale, J. Dongarra, G. Bosilca, A. Aiken, H. Jordan, K. Kennedy, E. Bueler, R. Ferreira, M. Frigo, W. Carlson, N. Carriero, I. Chivers, H. Kaiser и другими.

Средства автоматизации конструирования параллельных программ повышают практическую доступность вычислительных ресурсов, переносимость программ, сокращают время разработки, отладки и модификации программ, снижают требования к квалификации программиста.

Таким образом, автоматизация конструирования параллельных программ численного моделирования для мультикомпьютеров является актуальной темой.

**Целью** диссертационного исследования является разработка экспериментальной системы автоматического конструирования параллельных программ численного моделирования для мультикомпьютеров на основе теории структурного синтеза параллельных программ на вычислительных моделях. **Задачи**:

* разработка языка конструирования параллельных программ LuNA (Language for Numerical Algorithms),
* разработка алгоритмов трансляции и исполнения LuNA-программ,
* реализация разработанных системных алгоритмов в виде экспериментальной системы конструирования параллельных программ LuNA,
* экспериментальное исследование нефункциональных свойств системы LuNA и конструируемых программ.

**Методология и методы исследования.** Проведённое исследование базируется на результатах теории алгоритмов и математической логики. За основу при разработке модели фрагментированного алгоритма взято понятие вычислительной модели из теории синтеза параллельных программ на вычислительных моделях (В.А. Вальковский, В.Э. Малышкин). Результаты теории вычислимости использованы для доказательства свойства универсальности модели фрагментированного алгоритма. При разработке алгоритмов использовались результаты теории алгоритмов, в том числе методы разработки параллельных программ в общей и распределённой памяти с учётом сложившихся практик разработки численных параллельных программ. При разработке архитектуры и программного кода системы LuNA применялись методы и сложившиеся практики разработки программного обеспечения, в частности, методика объектно-ориентированного программирования. При разработке транслятора использовались результаты теории компиляции.

**Положения, выносимые на защиту:**

* Модель фрагментированного алгоритма,
* Язык LuNA описания фрагментированных алгоритмов,
* Системные алгоритмы трансляции и распределённого исполнения фрагментированных алгоритмов, описанных на языке LuNA,
* Проект системы LuNA автоматического конструирования параллельных программ для мультикомпьютеров.

**Степень достоверности и апробация результатов**

Достоверность полученных результатов обеспечивается применением использованных методов исследования, а также подтверждается теоретическим анализом и результатами экспериментального исследования. Все основные результаты работы докладывались на конференциях, научных семинарах и опубликованы в профильных рецензируемых печатных изданиях.

**Научной новизной** обладают следующие основные результаты работы:

* Модель фрагментированного алгоритма,
* Язык LuNA описания фрагментированных алгоритмов,
* Алгоритмы трансляции и распределённого исполнения фрагментированных алгоритмов:
  + базовый алгоритм распределённой интерпретации фрагментированных алгоритмов,
  + алгоритм Rope of Beads динамического отображения фрагментированных алгоритмов на узлы мультикомпьютера,
  + динамический алгоритм распределённой сборки мусора,
  + алгоритм распределённого обнаружения остановки системы

**Теоретическая значимость работы.** Предложенная модель фрагментированного алгоритма позволяет ставить задачу конструирования распределённой программы, соответствующей заданной функциональной спецификации, предназначенной для исполнения на мультикомпьютере с заданной конфигурацией и обладающей требуемыми нефункциональными свойствами.

На основе разработанной системы LuNA возможно экспериментальное исследование различных системных алгоритмов конструирования и исполнения параллельных программ (алгоритмов управления распределением ресурсов, планирования вычислений, балансировки вычислительной нагрузки по узлам мультикомпьютера и т.п.).

**Практическая значимость работы.** Разработанная система LuNA упрощает разработку, отладку и модификацию параллельных программ численного моделирования на мультикомпьютерах за счёт автоматизации программирования, в том числе отладки коммуникаций, синхронизации процессов и потоков, распределения и динамического перераспределения данных и вычислений по узлам мультикомпьютера, сборки мусора и пр.

**Личный вклад соискателя** заключается в обсуждении постановки задачи, разработке модели фрагментированного алгоритма на основе существующего понятия вычислительной модели, разработке языка LuNA описания фрагментированных алгоритмов, разработке системных алгоритмов трансляции и исполнения фрагментированных алгоритмов, представленных на языке LuNA, проектировании системы LuNA и разработке всех её основных компонентов. Все выносимые на защиту результаты получены автором лично.

**Публикации.** Основные результаты по теме диссертации изложены в 30 публикациях, из которых 21 опубликованы в журналах, рекомендованных ВАК (из них 7 индексированы в Web of Science, 16 — в Scopus); 26 — в тезисах докладов.

**Объём и структура работы.** Диссертация состоит из введения, 3 глав и заключения. Объём диссертации — 137 страниц, 15 рисунков, 5 листингов. Список литературы содержит 109 наименований.

**Содержание работы**

**Во введении** обосновывается актуальность исследования, теоретическая и практическая значимость полученных результатов, формулируется цель и задачи работы, выносимые на защиту положения, даётся краткая характеристика структуры диссертации.

**В главе 1** представлен обзор и анализ литературы по теме автоматического конструирования параллельных программ для мультикомпьютеров.

В основе диссертационного исследования лежат работы Ю.И. Янова, В.Е. Котова, В.А. Вальковского и В.Э. Малышкина, в которых задача конструирования программы рассматривается как «свёртка» рекурсивно-перечислимого множества функциональных термов, описывающих прикладной алгоритм, в конечные аппаратные ресурсы. При этом множество термов описывает не отдельные числа и операции над ними, а некоторые крупноблочные программные модули, накопленные в той или иной предметной области (сборочное программирование). Родственная тема развивалась Э.Х. Тыугу при существенном вкладе А.Л. Фуксмана по созданию языка Утопист. Также проблематика разрабатывалась И.Б. Задыхайло и А.Н. Андриановым в рамках проекта Норма. В.А. Крюков разрабатывал подход к конструированию параллельных программ на основе последовательного кода. С.М. Абрамовым и А.А. Московским была разработана Т-Система.

За рубежом проблематика разрабатывалась авторами C. Hoare, L. Lamport, L. Kale (система Charm++), J. Dongarra и G. Bosilca (проекты PaRSEC и DPLASMA), A. Aiken (системы Legion и Regent), H. Jordan (система AllScale), K. Kennedy (HPF), E. Bueler (PETSc), R. Ferreira (система Anthill), M. Frigo (язык Cilk), W. Carlson (язык UPC), N. Carriero (система Linda), I. Chivers (Co-Array Fortran), H. Kaiser (система HPX) и другими.

Анализ литературы показывает, что существующие подходы к конструированию параллельных программ являются либо относительно узкоспециализированными, либо не обеспечивают высокой эффективности конструируемых программ для достаточно широкого класса приложений. Выделяется подход с управляемым исполнением, когда система предназначена для широкого класса приложений, но помимо собственно описания алгоритма от программиста требуется конкретизовать способ его отображения на ресурсы мультикомпьютера во времени. К таким системам относятся PaRSEC, Legion и AllScale. Также анализ показывает, что проблема автоматического конструирования эффективных параллельных программ далеко не решена и требует дальнейшей проработки.

**В главе 2** представлены теоретические результаты работы, обеспечивающие создание требуемой системы: модель фрагментированного алгоритма (ФА), лежащая в основе входного языка системы, и системные алгоритмы, которые обеспечивают конструирование и распределённое выполнение параллельной программы, реализующей заданный ФА.

За основу ФА как входного представления системы взято понятие вычислительной модели (ВМ) из теории структурного синтеза (В.А. Вальковский, В.Э. Малышкин).

ФА по существу является описанием множества информационно-зависимых задач (фрагментов вычислений, ФВ), каждая из которых представляет собой последовательную процедуру без побочных эффектов, а информационные связи между задачами явно выражены в виде перечня входных и выходных переменных задачи (фрагментов данных, ФД). В работе предложен формализм, позволяющий описывать ФА на основе трёх операторов — исполнения, арифметического цикла и цикла с предусловием, а также определено исполнение ФА как процесс вычисления значений одних ФД из других путём исполнения ФВ в порядке, не противоречащем информационным зависимостям.

ФА является универсальным (полным по Тьюрингу) представлением, что доказывается в диссертации путём повторения определения рекурсивной функции в терминах ФА.

ФА удовлетворяет разработанным требованиям ко входному языку системы — оно явно параллельно, крупноблочно, не привязано к вычислителю, может быть различным образом отображено на аппаратные ресурсы.

Далее в главе представлены алгоритмы транслятора и исполнительной системы, обеспечивающие распределённую реализацию ФА на мультикомпьютерах.

Сначала вводится понятие локационной функции, которая каждому ФВ и ФД по их идентификатору ставит в соответствие вычислительный узел, в зависимости от времени путём определения для каждого вычислительного узла номер соседнего узла, где следует продолжить поиск искомого фрагмента (ФВ или ФД). Таким образом задаётся последовательность вычислительных узлов, которая начинается на узле, где требуется найти фрагмент, а заканчивается на узле, где этот фрагмент должен находиться. Понятие локационной функция является ключевым в распределённой реализации ФА, т.к., с одной стороны, позволяет осуществлять доставку значений ФД с узла хранения на узлы потребления, и, с другой стороны, управлять распределением и динамическим перераспределением фрагментов по узлам мультикомпьютера.

Предложен *базовый распределённый алгоритм исполнения ФА*, который осуществляет исполнение ФА путём исполнения всех ФВ по готовности значений их входных ФД. При этом ФД доставляются с узлов, на которых они были вычислены, на узлы потребления. Алгоритм децентрализован и использует локационную функцию для распределения и поиска фрагментов.

*Алгоритм Rope of Beads* является распределённым децентрализованным алгоритмом распределения и перераспределения фрагментов по узлам мультикомпьютера, и представляет собой возможную реализацию локационной функции. Алгоритм экономит коммуникации, совместим с разными алгоритмами динамической балансировки нагрузки на узлы, обладает малыми накладными расходами, полностью децентрализован, коммуникации осуществляются только между соседними узлами мультикомпьютера.

Также представлено два децентрализованных *алгоритма доставки ФД* (к потребляющим их ФВ), три дополняющих друг друга *алгоритма распределённой сборки мусора* (удаления ФД, которые более не понадобятся). *Алгоритм оптимизации исполнения ФА на основе профилирования* позволяет корректировать отображение фрагментов на узлы на основе анализа профиля предыдущего исполнения ФА за счёт выявления простаивающих узлов и перераспределения на них части фрагментов с перегруженных узлов. Предложена модификация алгоритма Дейкстры—Шольтена для распределённого *обнаружения момента окончания исполнения ФА*. Предложен ч*астный алгоритм исполнения ФА на основе трассировочной информации* о предыдущем исполнении ФА, обладающий меньшими накладными расходами.

**В главе 3** рассматриваются вопросы программной реализации предложенных системных алгоритмов в виде системы программирования LuNA. Структурно система состоит из двух основных компонентов — компилятора и исполнительной системы.

*Предложен язык* LuNA (Language for Numerical Algorithms), основанный на предложенной модели ФА и расширяющий её дополнительными средствами, повышающими удобство использования («синтаксический сахар»). Язык соответствует предъявляемым к нему требованиям — он близко следует модели ФА, что делает его простым для использования неспециалистом, является текстовым, что позволяет использовать стандартные средства разработки, модули описываются на распространённом и высокопроизводительном языке C++, имеет средства модульности, C-подобный синтаксис для простоты освоения, простую грамматику.

*Предложено* *исполняемое представление LuNA-программы* (ИП) на основе мультиагентного подхода, где каждый агент реализует некоторый ФВ и управляется императивной программой (в противовес декларативному языку LuNA) генерируемой на языке C++. ИП позволяет LuNA-компилятору вкладывать в него императивное управление и частично определять распределение ФВ и ФД по узлам, сохраняя при этом фрагментированную структуру программы во время исполнения для поддержки динамических свойств и обеспечения возможности профилирования и трассировки исполнения ФА. Использование языка C++ позволяет анализировать и оптимизировать ИП развитым инструментарием C++ разработки.

*Разработаны необходимые форматы представления данных.* Внутреннее представление LuNA-программы на промежуточных этапах компиляции разработано на основе распространённого и хорошо инструментально поддержанного формата JSON. Разработан формат (шаблонная структура) программы агента в ИП. Программа агента состоит из набора C++-функций, логически представляющих собой единую управляющую программу агента, но разбитую на части в тех местах, где агент может переходить в состояние длительного ожидания (входных ФД, например) или мигрировать на другой узел. Формат учитывает ограничения современных компиляторов на максимальное количество процедур в файле путём разбиения ИП на несколько файлов. Предложен формат представления скомпилированной программы в виде единственного файла динамической линковки (\*.so), объединяющего в себе программы агентов, JSON-представление ФА и некоторые сервисные функции. Также предложен формат хранения позиционной информации, позволяющий в сообщениях об ошибках указывать положения объектов в исходных листингах, а не в промежуточных файлах разных этапов компиляции, где ошибки были обнаружены.

*Рассмотрены ключевые вопросы организации системы LuNA.* Предложена организация исходного кода системы с учётом предъявляемых к ней требований расширяемости по системным алгоритмам и языку. Предложена организация однопроходного компилятора на основе классической для компиляторов конвейерной архитектуры, расширяемой по модулям компилятора и опирающейся на существующий развитый инструментарий. Предложена организация исполнительной системы как традиционной распределённой программы, работающей по принципу распределённого портфеля задач (ФВ) и распределённой базы данных (для хранения ФД), и совместимой с традиционной для высокопроизводительных мультикомпьютеров коммуникационным стандартом MPI. Рассмотрены вопросы сбора профилировочной информации, отладки LuNA-программ, а также возможности использования специализированных способов исполнения ФА (без исполнительной системы или с использованием частной исполнительной системы).

*Рассмотрены технические особеннсоти трансляции и исполнения LuNA-программ.* В частности, предложен алгоритм отслеживания значения имён при трансляции LuNA-программ с учётом областей видимости. Рассмотрены особенности поэтапного запрашивания входных ФД для случаев косвенной адресации. Представлена техника удалённых указателей, позволяющая сократить время поиска удалённого обработчика сообщений с логарифмического до константного. Рассмотрена возможность поддержки спецвычислителей (GPU) для реализации ФВ. Разобрана проблема больших листингов при реализации алгоритма воспроизведения трасс путём генерации распределённой C++ программы, предложено её решение путём использования простого интерпретатора трассы.

**В главе 4** представлены результаты экспериментального исследования разработанной системы LuNA.

Исследована работа предложенного алгоритма Rope of Beads для распределения и динамического перераспределения фрагментов по узлам применительно к задаче моделирования эволюции самогравитирующего протопланетного диска методом частиц-в-ячейках. Эксперимент показал, что динамическая балансировка нагрузки была обеспечена и привела к существенному уменьшению времени вычислений.

Исследована применимость системы LuNA к решению практических задач в совместных работах с коллегами из КазНУ (г. Алма-Ата, Казахстан), которые использовали систему LuNA для реализации численных алгоритмов в геофизических задачах фильтрации. Представлены экспериментальные исследования нескольких приложений в сравнении с ручной реализацией тех же алгоритмов. Эксперименты показали работоспособность системы. Эффективность системы LuNA на этих задачах уступала эффективности ручных реализаций до 1–2 порядков (по времени выполнения), причём отставание по эффективности сокращалось по мере развития системы (совместные работы велись несколько лет) и по мере перехода к задачам большего размера.

Экспериментально исследована возможность повышения эффективности исполнения LuNA-программ на основе профилирования на модельной задаче умножения плотных матриц, где была достигнута эффективность равная ручной реализации.

Экспериментально подтверждена возможность автоматического использования GPU для реализации ФВ на примере модельной задаче, реализующей явную конечно-разностную схему. Показано, что минимальное время достигается при совместном использовании GPU и CPU для определённых параметров задачи (отношение объёма вычислений к единице данных).

Экспериментально исследована эффективность алгоритма воспроизведения трасс в сравнении с обычным исполнением LuNA-программ на том же приложении метода частиц-в-ячейках. Показано существенное улучшение эффективности (уменьшение времени вычислений примерно на порядок).

**В заключении** сформулированы основные результаты работы: предложена модель фрагментированного алгоритма и язык LuNA описания фрагментированных алгоритмов, разработаны необходимые алгоритмы трансляции и исполнения LuNA-программ, спроектирована, реализована и экспериментально исследована система автоматического конструирования параллельных программ LuNA.

**Публикации автора по теме диссертации:**

*Статьи в журналах, включенных в Перечень рецензируемых научных изданий, в которых должны быть опубликованы основные научные результаты диссертаций на соискание ученой степени кандидата наук:*

1. V.E. Malyshkin, V.A. Perepelkin, A.A. Tkacheva. Control Flow Usage to Improve Performance of Fragmented Programs Execution // In Proc 13th International Conference on Parallel Computing Technologies. LNCS 9251. Springer, 2015. pp. 86-90. DOI: 10.1007/978-3-319-21909-7\_9. (Web of Science)

2. Victor E. Malyshkin, Vladislav A. Perepelkin, Georgy A. Schukin. Distributed Algorithm of Data Allocation in the Fragmented Programming System LuNA // In Proc 13th International Conference on Parallel Computing Technologies. LNCS 9251. Springer, 2015. pp. 80-85. DOI: 10.1007/978-3-319-21909-7\_8. (Web of Science)

3. Akhmed-Zaki D.Zh., Lebedev D.V., Perepelkin V.A. Implementation of a Three-Phase Fluid Flow (“Oil-Water-Gas”) Numerical Model in the LuNA Fragmented Programming System // In Proc 13th International Conference on Parallel Computing Technologies. LNCS 9251. Springer, 2015. pp. 489-497. DOI: 10.1007/978-3-319-21909-7\_47. (Web of Science)

4. Victor E. Malyshkin, Vladislav A. Perepelkin. The PIC Implementation in LuNA System of Fragmented Programming // The Journal of Supercomputing, Special Issue on Parallel Computing Technologies. Springer, 2014. pp. 89-97. DOI: 10.1007/s11227-014-1216-8. (Web of Science)

5. Victor Malyshkin, Vladislav Perepelkin. Optimization methods of parallel execution of numerical programs in the Luna fragmented programming system // The Journal of Supercomputing, Special issue on Enabling Technologies for Programming Extreme Scale Systems, Volume 61, Number 1 (2012). Springer, 2012. pp. 235-248. DOI: 10.1007/s11227-011-0649-6. (Web of Science)

6. Victor E. Malyshkin, Vladislav A. Perepelkin. LuNA Fragmented Programming System, Main Functions and Peculiarities of Run-Time Subsystem // Parallel Computing Technologies. 11th International Conference, PaCT 2011, Proceedings. LNCS 6873. Springer, 2011. pp. 53-61. (Web of Science)

7. Victor Malyshkin, Vladislav Perepelkin. Optimization of Parallel Execution of Numerical Programs in Luna Fragmented Programming System // Methods and Tools of Parallel Programming Multicomputers, Second Russia-Taiwan Symposium, MTPP 2010. LNCS 6083. Springer, 2010. pp. 1-10. (Web of Science)

8. Malyshkin, V., Akhmed-Zaki, D., Perepelkin, V. Parallel programs execution optimization using behavior control in LuNA system // J Supercomput. 2021. pp. 9771-9779. DOI: 10.1007/s11227-021-03654-2. (Scopus)

9. Malyshkin V., Perepelkin V. Trace-Based Optimization of Fragmented Programs Execution in LuNA System // Parallel Computing Technologies. PaCT 2021. Lecture Notes in Computer Science, vol 12942. Springer, Cham, 2021. pp. 3-10. DOI: 10.1007/978-3-030-86359-3\_1. (Scopus)

10. Belyaev N., Perepelkin V. High-Efficiency Specialized Support for Dense Linear Algebra Arithmetic in LuNA System // Parallel Computing Technologies. PaCT 2021. Lecture Notes in Computer Science, vol 12942. Springer, Cham, 2021. pp. 143-150. DOI: 10.1007/978-3-030-86359-3\_11. (Scopus)

11. Akhmed-Zaki, D., Lebedev, D., Perepelkin, V. Implementation of a 3D model automated heat equation using fragmented programming technology // J Supercomput. 2019. pp. 7827-7832. DOI: 10.1007/s11227-018-2710-1. (Scopus)

12. Akhmed-Zaki, D., Lebedev, D., Malyshkin, V., Perepelkin, V. Automated construction of high performance distributed programs in LuNA system // 15th International Conference on Parallel Computing Technologies, PaCT 2019; Almaty; Kazakhstan. LNCS 11657. Springer, 2019. pp. 3-9. DOI: 10.1007/978-3-030-25636-4\_1. (Scopus)

13. B. Daribayev, V. Perepelkin, D. Lebedev, D. Akhmed-Zaki. Implementation of the Two-Dimensional Elliptic Equation Model in LuNA Fragmented Programming System // 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT). 2018. pp. 1-4. (Scopus)

14. Malyshkin. V., Perepelkin. V., Schukin G. Scalable Distributed Data Allocation in LuNA Fragmented Programming System // Journal of Supercomputing, S.I.: Parallel Computing Technologies - 2017. Springer, 2017. pp. 1-7. DOI: 10.1007/s11227-016-1781-0. (Scopus)

15. Akhmed-Zaki, D., Lebedev, D., Perepelkin, V. Implementation of a three dimensional three-phase fluid flow (“oil–water–gas”) numerical model in LuNA fragmented programming system // Journal of Supercomputing (2017). - 73(2). Springer, 2017. pp. 624-630. DOI: 10.1007/s11227-016-1780-1. (Scopus)

16. Nikolay B., Perepelkin. V. Automated GPU Support in LuNA Fragmented Programming System // Parallel Computing Technologies. PaCT 2017. Lecture Notes in Computer Science, vol 10421.. Springer, Cham, 2017. pp. 272-277. DOI: 10.1007/978-3-319-62932-2\_26. (Scopus)

17. Малышкин В.Э., Перепёлкин В.А., Щукин Г.А. Распределённый алгоритм управления данными в системе фрагментированного программирования LuNA // Проблемы информатики. No 1(34). 2017. с. 78-88.

18. Перепёлкин В.А., Софронов И.В., Ткачёва А.А. Автоматизация конструирования численных параллельных программ с заданными нефункциональными свойствами на базе вычислительных моделей // Проблемы информатики. No 4(37). ИВМиМГ СО РАН (Новосибирск), 2017. с. 47-60.

19. Ажбаков А.А., Перепёлкин В.А. Разработка и реализация переносимых алгоритмов распределенного исполнения фрагментированных программ на неоднородных вычислителях // Проблемы информатики. No 1(42). ИВМиМГ СО РАН (Новосибирск), 2019. с. 51-69.

20. Д. Ахмед-Заки, Д. Лебедев, В. Малышкин, В. Перепелкин. Автоматизация конструирования распределенных программ численного моделирования в системе LuNA на примере модельной задачи // Проблемы информатики. No 4(45). ИВМиМГ СО РАН (Новосибирск), 2019. с. 53-64.

21. В.А. Перепелкин, И.И. Сумбатянц. Стенд для отладки и тестирования качества работы локальных системных распределенных алгоритмов динамической балансировки нагрузки // Вестник Южно-Уральского Государственного Университета, секция «Вычислительная математика и информатика». — том 4, № 3 (2015). 2015. с. 55-66.

*Публикации в других научных изданиях:*

1. В.А. Перепелкин. Компилятор с языка представления фрагментированных программ // Труды конференции молодых ученых. Новосибирск: ИВМиМГ СО РАН, 2009. с. 136-141.

2. V.E. Malyshkin, V.A. Perepelkin, V.K. Grankin, A.V. Talnikov. Automatic Provision of Dynamic Load Balancing in Fragmented Programs // Proceedings of the 12th International Conference on Humans and Computers. Aizu, Japan, 2009. pp. 230-234.

3. В.А. Перепелкин. Представление алгоритмов в технологии фрагментированного программирования // Труды Пятой Сибирской конференции по параллельным и высокопроизводительным вычислениям. Под ред. проф. А.В. Старченко. Томск: Изд-во Том. Ун-та, 2010. с. 149-152.

4. В.А. Перепелкин. Проблема распределения ресурсов мультикомпьютера в технологии фрагментированного программирования // Научный сервис в сети Интернет: поиск новых решений: Труды Международной суперкомпьютерной конференции (17-22 сентября 2012 г., г. Новороссийск). М.: Изд-во МГУ, 2012. с. 398-401.

5. Д.В. Лебедев, В.А. Перепелкин. Реализация одномерной краевой задачи нефть-вода в системе фрагментированного программирования LuNA // Материалы XIV Международной конференция "Высокопроизводительные параллельные вычисления на кластерных системах". ПНИПУ, г. Пермь, 2014.

6. Д.В. Лебедев, В.А. Перепелкин. Численное решение одномерной краевой задачи фильтрации жидкости для системы "нефть-вода" и ее реализация в системе фрагментированного программирования LuNA // Вестник Казахского национального университета им. Аль-Фараби, серия математика, механика информатика. No 3(82). 2014. с. 64-73.

7. Малышкин В.Э., Перепелкин В.А. Реализация метода частиц-в-ячейках в системе фрагментированного программирования LuNA // Научный сервис в сети Интернет: многообразие суперкомпьютерных миров Труды Международной суперкомпьютерной конференции. Российская академия наук Суперкомпьютерный консорциум университетов России. 2014. с. 328-334.

8. Перепелкин В.А., Сумбатянц И.И. Стенд для отладки и тестирования качества работы локальных системных распределенных алгоритмов динамической балансировки нагрузки // Параллельные вычислительные технологии (ПаВТ'2015) Труды международной научной конференции. 2015. с. 448-455.

9. В.Э. Малышкин, В.А. Перепёлкин. Автоматизация программирования в системе конструирования распределённых численных программ LuNA // Измерения, автоматизация и моделирование в промышленности и научных исследованиях (ИАМП-2018): материалы XIII Всероссийской научно-технической конференции студентов, аспирантов и молодых учёных с международным участием, 1-3 ноября 2018 г.. Алт. гос. техн. ун-т, БТИ. - Бийск: Изд-во Алт. гос. техн. ун-та, 2018. с. 524-527.