

На правах рукописи



Перепёлкин Владислав Александрович

СИСТЕМА LUNA АВТОМАТИЧЕСКОГО КОНСТРУИРОВАНИЯ
ПАРАЛЛЕЛЬНЫХ ПРОГРАММ ЧИСЛЕННОГО
МОДЕЛИРОВАНИЯ НА МУЛЬТИКОМПЬЮТЕРАХ

Специальность 2.3.5 — математическое и программное обеспечение
вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Новосибирск — 2022

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: **Легалов Александр Иванович**,
доктор технических наук, профессор,
Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет
«Высшая школа экономики»,
профессор Департамента программной инженерии
Факультета компьютерных наук

Курносов Михаил Георгиевич,
доктор технических наук, профессор,
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Сибирский государственный университет
телекоммуникаций и информатики»,
профессор Кафедры вычислительных систем

Ведущая организация: Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Нижегородский
государственный университет им. Н.И. Лобачевского»

Защита состоится «21» февраля 2023 г. в 15:00 на заседании диссертационного совета 24.1.047.01 на базе Федерального государственного бюджетного учреждения науки Института вычислительной математики и математической геофизики Сибирского отделения Российской академии (ИВМиМГ СО РАН) наук по адресу: 630090, г. Новосибирск, проспект академика Лаврентьева, 6, конференц-зал ИВМиМГ СО РАН, тел. (383) 330-71-59.

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

Автореферат разослан «25» ноября 2022 г.

Ученый секретарь
диссертационного совета 24.1.047.01,
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 по количеству вычислительных узлов и использовать все доступные ресурсы вычислительной системы.

Решение этого круга проблем требует значительных трудозатрат и специальной квалификации, отличной от квалификации в той предметной области, к которой относится программа. Это обуславливает актуальность создания и применения средств автоматического конструирования численных параллельных программ с требуемыми свойствами, т.к. такие средства способны взять на себя решение обозначенных проблем хотя бы частично и тем самым повысить практическую доступность вычислительных ресурсов, переносимость программ, сократить время разработки, отладки и модификации программ, снизить требования к квалификации программиста.

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. Тыгу, 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 С. 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 др.

Настоящая работа вносит вклад в развитие подхода к автоматическому конструированию параллельных программ на вычислительных моделях, предложенный В.А. Вальковским и В.Э. Малышкиным.

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

Для достижения поставленной цели в работе ставятся и решаются следующие **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 ею программ.

Методология и методы исследования. 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. 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.

Научная новизна работы:

- на основе существующего понятия вычислительной модели предложена модель фрагментированного алгоритма;
- разработан язык LuNA описания фрагментированных алгоритмов;
- разработаны алгоритмы, обеспечивающие трансляцию и распределённое исполнение фрагментированных алгоритмов:
 - базовый алгоритм распределённой интерпретации фрагментированных алгоритмов,
 - алгоритм Rope of Beads динамического отображения фрагментированных алгоритмов на узлы мультимпьютера,
 - 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 мультимпьютере 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 мультимпьютера и т.п.).

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

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 мультимикросистем.

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.

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 27 статьях, из которых 22 опубликованы в журналах, рекомендованных ВАК (7 индексируются в Web of Science, 16 — в Scopus). Также опубликовано 26 работ в виде тезисов докладов.

Объём и структура работы. Диссертация состоит из введения, 4 глав и заключения. Объём диссертации — 118 страниц без приложений (с приложениями — 135 с.), 16 рисунков, 2 таблицы и 15 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 мультимикомпьютеров.

В работах Ю.И. Янова, В.Е. Котова, В.А. Вальковского и В.Э. Малышкина задача конструирования программы рассматривается как «свёртка» рекурсивно-перечислимое множества функциональных термов, описывающих прикладной алгоритм, в программу для конечного объёма аппаратных ресурсов. При этом термы описывают не отдельные числа и операции над ними, а крупноблочные программные модули (сборочное программирование). Родственная тема развивалась Э.Х. Тыгу 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 T-Система.

За рубежом проблематика разрабатывалась авторами С. 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 управляемого исполнения, когда система предназначена для широкого класса приложений, но помимо собственно описания алгоритма от программиста требуется конкретизировать способ его отображения на ресурсы мультимикомпьютера 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, обеспечивающие 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 исполнение модуля (последовательной процедуры) без побочных эффектов, а информационные связи между задачами явно выражены через перечисление входных и выходных аргументов 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 по мере вычисления их входных ФД.

ФА является универсальным (полным по Тьюрингу) представлением алгоритмов, что доказывается в диссертации путём повторения определения рекурсивной функции в терминах ФА. Представление явно параллельно, крупноблочно, не привязано к вычислителю, может быть различным образом отобразено на аппаратные ресурсы, в т.ч. динамически.

Далее в главе представлены алгоритмы трансляции и распределённого исполнения ФА, обеспечивающие распределённую реализацию ФА на мультимпьютерах.

Для поиска ФВ и ФД в мультимпьютере вводится понятие *локационной функции*, которая каждому идентификатору ФВ и ФД динамически ставит в соответствие соседний вычислительный узел, где следует продолжить поиск. Применение локационной функции позволяет абстрагироваться от конкретных алгоритмов распределения и перераспределения ФВ и ФД по узлам мультимпьютера и позволяет осуществлять доставку значений ФД с узла выработки на узлы потребления.

Предложен *базовый распределённый алгоритм исполнения ФА*, который осуществляет исполнение ФА путём исполнения всех составляющих его 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 ФВ и ФД по узлам мультимпьютера и представляет собой вариант реализации локационной функции. Алгоритм учитывает топологию сети, совместим с разными алгоритмами динамической балансировки нагрузки на узлы, обладает малыми накладными расходами, полностью децентрализован, коммуникации осуществляются только между соседними узлами мультимпьютера для снижения нагрузки на сеть.

Предложено два децентрализованных дополняющих друг друга *алгоритма доставки ФД* (к потребляющим их ФВ), три дополняющих друг друга *алгоритма распределённой сборки мусора* (удаления ФД, которые более не понадобятся). Предложен *алгоритм оптимизации исполнения ФА на основе профилирования*, который позволяет корректировать отображение фрагментов на вычислительные 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 *частный алгоритм исполнения ФА на основе трассировочной информации* о предыдущем исполнении ФА, обладающий меньшими накладными расходами, чем исполнительная система.

В главе 3 разрешаются вопросы программной реализации предложенных системных алгоритмов в виде системы программирования LuNA. Структурно система состоит из двух основных компонентов — транслятора 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 прямо следует модели ФА, что делает его простым для использования неспециалистом, является текстовым, что позволяет использовать стандартные средства разработки, позволяет описывать модули (процедуры ФВ) 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 отличие от декларативного языка LuNA) на C++. ИП получается в результате трансляции LuNA-программы и является входным представлением для исполнительской системы. ИП позволяет LuNA-транслятору вкладывать в него императивное управление и частично определять распределение ФВ и ФД по узлам, сохраняя при этом фрагментированную структуру программы во время исполнения. Это, в свою очередь, позволяет исполнительской системе обеспечивать динамические свойства, профилировать и трассировать исполнение ФА. Использование языка C++ в ИП позволяет анализировать и оптимизировать ИП развитым инструментарием C++ разработки, в т.ч. автоматически.

Разработаны необходимые форматы представления данных. Внутреннее представление LuNA-программы на промежуточных этапах трансляции разработано на основе распространённого и хорошо инструментально поддержанного формата JSON. Разработана 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 оттранслированной программы в виде единственного файла динамической линковки (.so), объединяющего в себе программы агентов, JSON-представление ФА и некоторые сервисные процедуры. Также разработана каскадная схема хранения и обработки позиционной информации, определяющей исходный файл и позицию в нём описаний сущностей, существующих на различных этапах трансляции и исполнения для облегчения отладки и повышения информативности сообщений об ошибках.*

Рассмотрены ключевые вопросы организации системы LuNA. Предложена организация исходного кода системы с учётом требований

расширяемости системными алгоритмами и языку. Предложена организация однопроходного транслятора на основе классической для трансляторов конвейерной архитектуры, расширяемой модулями транслятора и опирающейся на существующий развитый инструментарий разработки трансляторов. 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 традиционным для высокопроизводительных мультимпьютеров коммуникационным стандартом MPI (но не привязанной к нему). 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 особенности 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, возникающая при реализации алгоритма воспроизведения трасс путём генерации распределённой C++ программы, предложено и реализовано её решение за счёт использования простого интерпретатора трассы.

В главе 4 представлены результаты экспериментального исследования разработанной системы LuNA на предмет её работоспособности и нефункциональных свойств конструируемых программ.

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 Казахского Национального Университета (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 автоматического повышения эффективности исполнения LuNA-программ на основе профилирования на модельной задаче умножения плотных матриц, где была достигнута производительность близкая к ручной реализации.

Экспериментально подтверждена возможность автоматического использования GPU для реализации ФВ на примере модельной задачи, реализующей явную конечно-разностную схему. При этом от пользователя системы требуется предоставить модуль (kernel) для GPU, а передача данных и организация совместной работы GPU и CPU обеспечивается автоматически. 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.

Главным результатом работы является то, что на основе теории структурного синтеза параллельных программ на вычислительных моделях построена система автоматического конструирования параллельных программ численного моделирования на мультикомпьютерах, которая способна обеспечить конструирование параллельных программ удовлетворительного качества, по крайней мере, в некоторых практически значимых предметных областях.

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. 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, МТПП 2010. — LNCS 6083. — Springer, 2010. — С. 1–10. (Web of Science)
2. 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. — С. 53–61. (Web of Science)
3. 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. — Vol. 61, No 1. — Springer, 2012. — С. 235-248. — DOI: 10.1007/s11227-011-0649-6. (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. — С. 89–97. — DOI 10.1007/s11227-014-1216-8. (Web of Science)
5. V.E. Malyshkin, V.A. Perepelkin, A.A. Tkacheva. Control Flow Usage to Improve Performance of Fragmented Programs Execution // Proc 13th International Conference on Parallel Computing Technologies. LNCS. — Т.

9251. — Springer, 2015. — С. 86–90. — DOI 10.1007/978-3-319-21909-7_9. (Web of Science)
6. 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. — С. 80–85. (Web of Science)
7. 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 // Proc 13th International Conference on Parallel Computing Technologies. LNCS. — Т. 9251. — Springer, 2015. — С. 489–497. — DOI 10.1007/978-3-319-21909-7_47. (Web of Science)
8. В.А. Перепелкин, И.И. Сумбатьянц. Стенд для отладки и тестирования качества работы локальных системных распределенных алгоритмов динамической балансировки нагрузки // Вестник Южно-Уральского Государственного Университета, секция «Вычислительная математика и информатика». — Т. 4, № 3. — 2015. — С. 55–66.
9. 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. — С. 1–7. — DOI: 10.1007/s11227-016-1781-0. (Scopus)
10. 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. — № 73 (2). — Springer, 2017. — С. 624–630. — DOI 10.1007/s11227-016-1780-1. (Scopus)
11. Nikolay B., Perepelkin. V. Automated GPU Support in LuNA Fragmented Programming System // Parallel Computing Technologies (PaCT) 2017. Lecture Notes in Computer Science. — Т. 10421. — Springer, Cham, 2017. — С. 272–277. — DOI 10.1007/978-3-319-62932-2_26. (Scopus)
12. 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.
13. Перепёлкин В.А., Софронов И.В., Ткачёва А.А. Автоматизация конструирования численных параллельных программ с заданными нефункциональными свойствами на базе вычислительных моделей //

- Проблемы информатики. — № 4 (37). — ИВМиМГ СО РАН. — 2017. — С. 47–60.
14. B. Daribayev, V. Perepelkin, D. Lebedev [и др.]. 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. — С. 1–4. (Scopus)
15. Akhmed-Zaki, D., Lebedev, D., Perepelkin, V. Implementation of a 3D model heat equation using fragmented programming technology // J Supercomput. — 2019. — С. 7827–7832. — DOI 10.1007/s11227-018-2710-1. (Scopus)
16. 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. — С. 3–9. — DOI 10.1007/978-3-030-25636-4_1. (Scopus)
17. Ажбаков А.А., Перепёлкин В.А. Разработка и реализация переносимых алгоритмов распределенного исполнения фрагментированных программ на неоднородных вычислителях // Проблемы информатики. — № 1 (42). — ИВМиМГ СО РАН. — 2019. — С. 51–69.
18. 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.
19. Перепелкин В.А. Система LuNA автоматического конструирования параллельных программ численного моделирования на мультикомпьютерах // Проблемы информатики. — No 1(46). — ИВМиМГ СО РАН (Новосибирск), 2020. — С. 66–74. — DOI: 10.24411/2073-0667-2020-10004.
20. Malyshkin, V., Akhmed-Zaki, D., Perepelkin, V. Parallel programs execution optimization using behavior control in LuNA system // J Supercomput. — Springer, 2021. — С. 9771-9779. — DOI: 10.1007/s11227-021-03654-2. (Scopus)
21. Malyshkin V., Perepelkin V. Trace-Based Optimization of Fragmented Programs Execution in LuNA System // Parallel Computing Technologies. PaCT 2021. Lecture Notes in Computer Science. — Т. 12942. — Springer, Cham, 2021. — DOI 10.1007/978-3-030-86359-3_1. (Scopus)

22. 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. — Т. 12942. — Springer Cham, 2021. — DOI 10.1007/978-3-030-86359-3_11. (Scopus)

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. — С. 230–234.

3. В.А. Перепелкин. Представление алгоритмов в технологии фрагментированного программирования // *Труды 5 Сибирской конференции по параллельным и высокопроизводительным вычислениям*. Под ред. проф. А.В. Старченко. — Томск: Изд-во Том. Ун-та, 2010. — С. 149–152.

4. 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*. — № 3 (82). — 2014. — С. 64–73.

5. 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 конф. РАН, Суперкомпьютерный консорциум университетов России*. — 2014. — С. 328–334.