

Учебно-исследовательские системы для изучения параллельных методов

В.П. Гергель, Е.А. Козин, А.В. Линева, А.А. Штанюк

Нижегородский государственный университет им. Н.И. Лобачевского

Подготовка кадров в области суперкомпьютерных вычислений является важной задачей. Вместе с тем обучение является крайне сложным. Большое разнообразие вычислительных систем и технологий усложняют обучение [3]. Помочь в обучении могут учебно-исследовательские системы. Мы представляем два примера подобных систем ParaLab[1, 5] и ParaLib[8]. В системах возможно выбрать из заданного списка решаемую задачу. Для задач могут быть изучены методы реализации параллельных алгоритмов. При заданном методе и различных начальных условиях могут быть выполнены вычислительные эксперименты. ParaLab позволяет наглядно отобразить решение задачи на смоделированной вычислительной системе, а так же получить оценки производительности и эффективности алгоритмов. ParaLib демонстрирует возможность использования различных современных технологий параллельного программирования и параллельных языков для решения поставленных задач (например, OpenMP, MPI, Co-Array Fortran[6] и Chapel[7]). Обе системы позволяют накапливать результаты для последующего анализа. Открытая архитектура систем позволяет слушателям добавлять собственные реализации алгоритмов. Программные системы ParaLab и ParaLib используются при обучении студентов в Нижегородском университете.

1. Введение

Обучение методам параллельного программирования является важной задачей современного ИТ-образования. Для формирования необходимых компетенций, необходимо понимать работу алгоритмов распараллеливания и оценивать (как количественно так и качественно) затраты вычислительных ресурсов. Большую пользу в этом направлении может принести использование в учебном процессе специализированных учебно-исследовательских систем.

В рамках статьи рассматриваются две программные системы ParaLab и ParaLib, созданные и используемые в ННГУ им. Н.И.Лобачевского в рамках курсов [4], посвященных параллельному программированию и суперкомпьютерным технологиям.

Система ParaLab представляет собой интегрированную среду для изучения и исследования параллельных алгоритмов решения для большинства «классических» вычислительных задач (матричные вычисления, сортировка, обработка графов, оптимизация и др.) [1]. Широкий набор имеющихся средств визуализации процесса выполнения эксперимента и анализа полученных результатов позволяет изучить эффективность использования тех или иных алгоритмов на разных вычислительных системах, сделать выводы о масштабируемости алгоритмов и определить возможное ускорение процесса параллельных вычислений.

Система может быть использована для организации лабораторного практикума по учебным курсам в области параллельного программирования, в рамках которого обеспечивается возможность:

- моделирования многопроцессорных вычислительных систем с различной топологией сети передачи данных;
- получения визуального представления о вычислительных процессах и операциях передачи данных, происходящих при параллельном решении разных вычислительных задач;
- построения оценок эффективности изучаемых методов параллельных вычислений.

Проведение такого практикума может быть организовано на обычных однопроцессорных компьютерах под управлением операционных систем семейства Windows. Для исполнения алгоритмов используется режим имитации параллельных вычислений.

Библиотека параллельных алгоритмов ParaLib предназначена для изучения и освоения современных технологий и языков параллельного программирования на примерах реализации

параллельных алгоритмов для большого числа задач вычислительной математики. На примере представленных в системе вычислительных задач и параллельных методов их решения можно сравнить сложность программной реализации параллельных алгоритмов и эффективность разработанных параллельных программ. Библиотека дополнена учебно-исследовательской системой, которая позволяет:

- выбрать вычислительную задачу;
- выбрать последовательный/параллельный метод для решения выбранной задачи;
- выбрать язык/технологии параллельного программирования для решения выбранной задачи;
- задать параметры задачи;
- определить число используемых процессоров;
- выполнить эксперимент, в ходе которого будет выполнен запуск решения выбранной задачи;
- накапливать и анализировать результаты выполненных экспериментов.

Выполнение экспериментов может осуществляться на локальном компьютере или на вычислительной системе с распределенной памятью. Программные реализации параллельных методов выполнены с использованием разных языков и технологий параллельного программирования, в числе которых:

- технология параллельного программирования для систем с общей памятью OpenMP;
- технология параллельного программирования для систем с распределенной памятью MPI;
- язык параллельного программирования Chapel (Cascade High Productivity Language);
- язык параллельного программирования Co-Array Fortran.

Все реализации выполнены для операционных систем семейства Windows.

Системы ParaLab и ParaLib органично дополняют друг друга, позволяя ставить различные вычислительные эксперименты, визуально изучать процессы вычислений и численно оценивать их параметры.

Далее приводится более полное описание рассматриваемых учебно-исследовательских систем.

2. Учебно-исследовательская система ParaLab

Учебно-исследовательская система ParaLab (*Параллельная Лаборатория*) представляет собой программный комплекс, который позволяет проводить как реальные параллельные вычисления на многопроцессорной вычислительной системе, так и имитировать такие эксперименты на одном последовательном компьютере с визуализацией процесса решения сложной вычислительной задачи.

При проведении имитационных экспериментов ParaLab предоставляет следующие возможности для пользователя:

1. Смоделировать параллельную вычислительную систему. В процессе моделирования возможно *определить топологию* параллельной вычислительной системы для проведения экспериментов (рис. 1), *задать число процессоров* в этой топологии, *установить производительность* процессоров, *выбрать характеристики коммуникационной среды* и *способ коммуникации*.

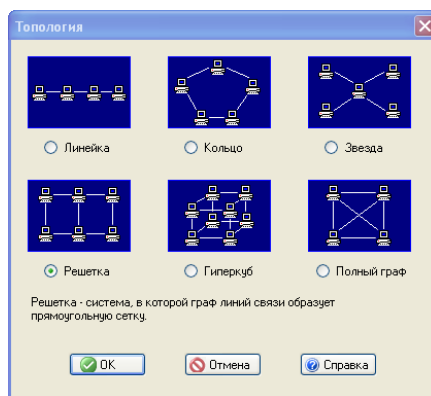


Рис. 1. Диалог выбора топологии системы

2. Задать условия вычислительного эксперимента. Для задания условий необходимо осуществить постановку вычислительной задачи, для которой в составе системы ParaLab имеются реализованные параллельные алгоритмы решения. Выполнить задание параметров задачи, а так же выбрать параллельный метод для решения выбранной задачи.

3. Устанавливать параметры визуализации. Система ParaLab позволяет установить параметры визуализации для выбора желаемого темпа демонстрации, способа отображения пересылаемых между процессорами данных, степени детальности визуализации выполняемых параллельных вычислений (рис. 2);

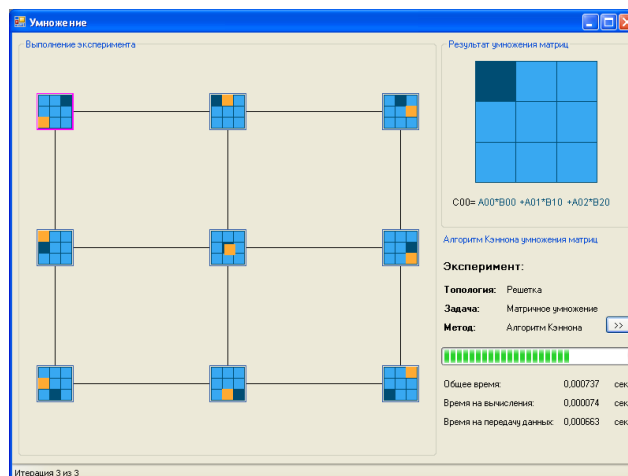


Рис. 2. Визуализация процесса параллельных вычислений при выполнении матричного умножения

4. Выполнять вычислительные эксперименты. В системе ParaLab можно выполнить эксперимент для параллельного решения выбранной задачи; при этом в системе ParaLab может быть сформировано несколько различных заданий для проведения экспериментов с отличающимися типами многопроцессорных систем, задач или методов параллельных вычислений, для которых выполнение эксперимента может происходить одновременно (в режиме разделения времени); одновременное выполнение эксперимента для нескольких заданий позволяет наглядно сравнивать динамику решения задачи различными методами, на разных топологиях, с разными параметрами исходной задачи. При выполнении серии экспериментов, требующих длительных вычислений, в системе имеется возможность их проведения в автоматическом режиме с запоминанием результатов в журнале экспериментов для организации последующего анализа полученных данных;

5. Накапливать и анализировать результаты вычислительных экспериментов. Система ParaLab позволяет накапливать и анализировать результаты выполненных экспериментов; по запомненным результатам в системе имеется возможность построения графиков, характеризующих параллельные вычисления зависимостей (времени решения, ускорения, эффективности) от параметров задачи и вычислительной системы (рис. 3).

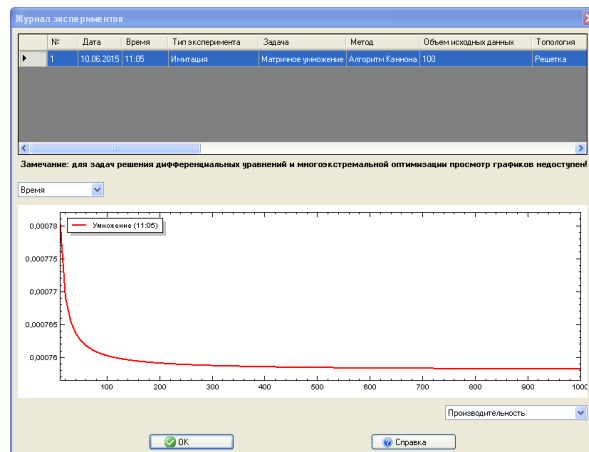


Рис. 3. Окно журнала эксперимента с результатами

Одной из важнейших характеристик системы является возможность выбора способов проведения экспериментов. Эксперимент может быть выполнен в *режиме имитации*, то есть проведен на одном процессоре без использования каких-либо специальных программных средств типа библиотек передачи сообщений. Кроме того, в рамках системы ParaLab обеспечивается возможность следующих способов проведения *реального вычислительного эксперимента*:

- на одном компьютере, где имеется библиотека передачи сообщений MPI (многопоточное выполнение эксперимента); для данной библиотеки имеются общедоступные реализации, которые могут быть получены в сети Интернет и установлены на компьютере под управлением операционных систем MS Windows,
- на реальной многопроцессорной кластерной вычислительной системе,
- в режиме удаленного доступа к вычислительному кластеру.

Если проводится реальный эксперимент на многопроцессорной вычислительной системе или в режиме удаленного доступа, система ParaLab предоставляет возможность выбора типов вычислительных узлов.

При построении зависимостей временных характеристик от параметров задачи и вычислительной системы для экспериментов, выполненных в режиме имитации, используются теоретические оценки в соответствии с имеющимися моделями параллельных вычислений [1-3]. Для реальных экспериментов на многопроцессорных вычислительных системах зависимости строятся по набору результатов проведенных вычислительных экспериментов. Любой из проведенных ранее экспериментов может быть восстановлен для повторного проведения. Кроме того, обеспечена возможность ведения журнала экспериментов с записью туда постановки задачи, параметров вычислительной системы и полученных результатов.

Лабораторные занятия, проводимые при использовании системы ParaLab, могут проводиться, например, по следующему плану:

- обучаемый решает определенную сложную вычислительную задачу при помощи нескольких параллельных методов на заданной им вычислительной системе, сопоставляет результаты и дает их интерпретацию в рамках теории построения параллельных алгоритмов;
- обучаемый самостоятельно конструирует средствами комплекса несколько вычислительных систем таким образом, чтобы при решении задачи продемонстрировать основные теоретические понятия;
- обучаемый самостоятельно формирует одну или несколько вычислительных систем и решает задачи при различных значениях параметров вычислительной системы, изучая тем самым влияние параметров на временные характеристики алгоритма;
- обучаемый проводит реальные вычислительные эксперименты в режиме удаленного доступа к кластеру и сравнивает результаты реальных и имитационных экспериментов.

Реализованные таким образом процессы изучения и исследований позволят освоить теоретические положения и помогут формированию представлений о методах построения параллельных алгоритмов, ориентированных на решение конкретных прикладных задач.

3. Библиотека параллельных численных методов ParaLib

Как уже было отмечено ранее, библиотека ParaLib предназначена для изучения и сравнительного сопоставления разных языков и технологий параллельного программирования. Для поддержки процессов изучения библиотека дополнена учебно-исследовательской системой (УИС), общая структура которой представлена на рис. 4.

Учебно-исследовательская система библиотеки ParaLib обеспечивает следующие возможности.

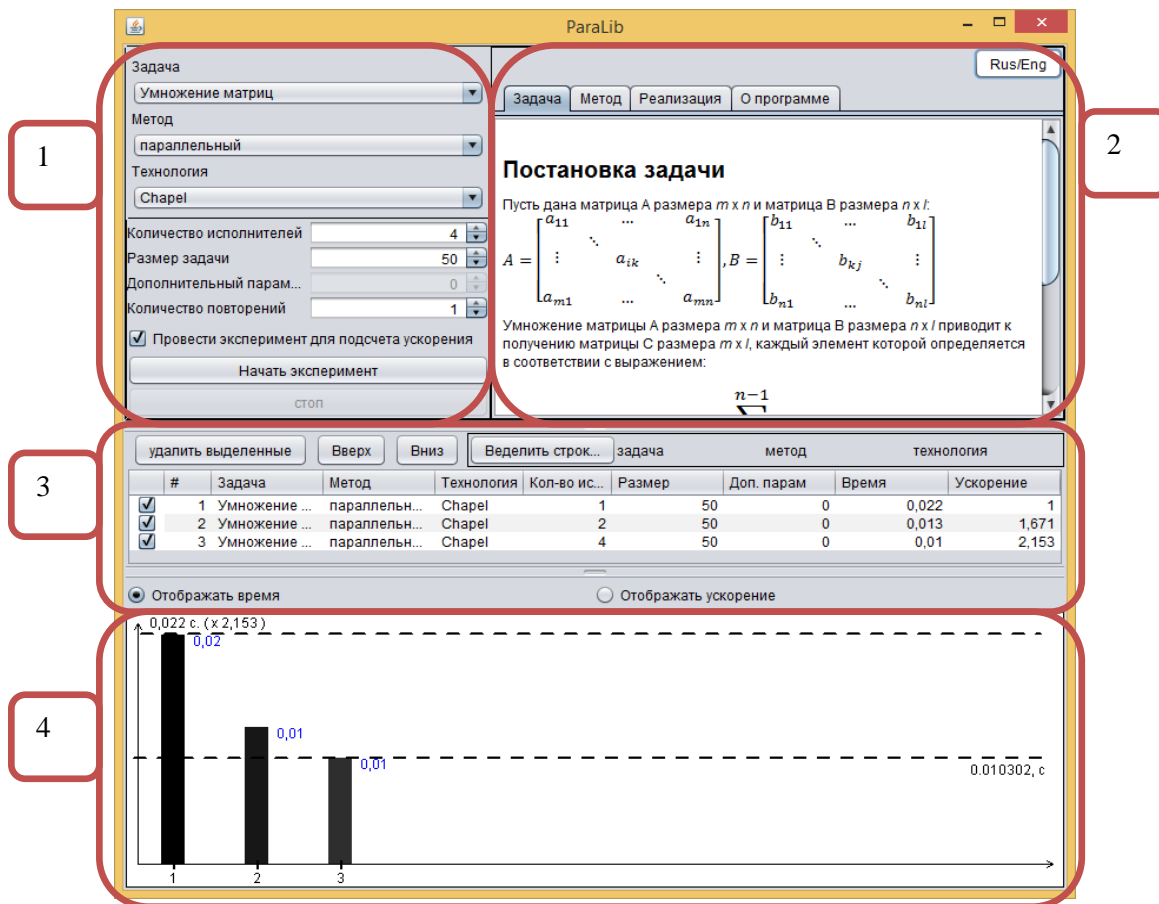


Рис. 4. Внешний вид учебно-исследовательской системы на основе библиотеки параллельных численных методов ParaLib

1. Выбор задачи и метода решения. Меню задания условий численного эксперимента располагается вверху слева диалогового окна системы (рис. 4 область 1).

Для проведения эксперимента необходимо вначале выбрать решаемую задачу. Выбрав задачу, можно указать метод ее решения. Определившись с алгоритмом, необходимо указать технологию или язык программирования, используемые в реализации.

В первой версии системы рассматривается ряд «классических» задач вычислительной математики, используемых для обучения основам параллельного программирования. Для задач представлены два способа реализации – последовательный и базовый параллельный алгоритмы. Все задачи содержат реализации с использованием технологий OpenMP и MPI. На примере задачи умножения матриц демонстрируется способ реализации и интеграции в библиотеку решений задач с применением языков параллельного программирования Co-Array Fortran и Chapel.

2. Получение контекстной информации. Информационная панель располагается вверху справа (рис.4 область 2) и позволяет получать необходимую информацию о выбранных алгоритмах и их реализациях. Выбор отображаемой информации осуществляется переключением вкладок панели.

Вкладка «Задача» используется для отображения постановки решаемой задачи. У каждой задачи может быть несколько методов решения. Описание выбранного метода находится в соответствующей вкладке. Выбрав задачу, метод реализации и технологию, можно изучить пример реализации алгоритма во вкладке «Реализация».

Последняя вкладка используется для отображения информации о библиотеке и реализованных в ней на текущий момент задачах и методах.

3. Выполнение вычислительного эксперимента. Выбрав алгоритм и интересующую реализацию, необходимо задать условия вычислительного эксперимента при помощи задания следующих параметров:

- Для эксперимента нужно указать количество исполнителей. В качестве исполнителей в зависимости от технологии или языка программирования могут выступать вычислительные потоки или процессы.

- Для эксперимента задается размер решаемой задачи. Размер интерпретируется в зависимости от задачи. Например, в задаче умножения матриц в качестве размера указывается размер матрицы, а задачах оптимизации на графах – количество вершин в графе. Предполагается, что описание значения параметра указано в постановке задачи (см. область 2 "Информационная панель").

- В ряде задач используется дополнительный параметр. Интерпретация значения зависит от постановки задачи или метода реализации. Подробную информацию о параметре так же можно узнать в постановке задачи или описании метода реализации. Параметр блокируется, если не используется.

- Количество повторений вычислительного эксперимента. Так как от запуска к запуску время выполнения может меняться, система предлагает возможность получения более достоверной оценки времени выполнения путем повторного запуска эксперимента несколько раз.

Система позволяет автоматически подсчитать ускорение для проведенных экспериментов. Для подсчета ускорения необходим запуск в один поток или процесс в зависимости от технологии или языка программирования. Подобный эксперимент проводится автоматически, если установлен выбор в поле «Провести эксперимент для подсчета ускорения».

Для запуска вычислительного эксперимента необходимо нажать на кнопку «Начать эксперимент». В любой момент времени эксперимент можно остановить, нажав кнопку остановки. На кнопке отображается таймер, указывающий на текущее время работы заданного эксперимента.

4. Сбор и анализ результатов вычислительных экспериментов. В середине диалогового окна (рис. 4 область 3) отображаются результаты всех выполненных экспериментов. Для каждого проведенного эксперимента в таблице указываются параметры эксперимента и время работы. В таблице слева можно поставить выбор тех запусков, которые следует отобразить на графике. По умолчанию время работы или ускорение отображается для всех выполненных экспериментов.

Для удобства пользователя реализован ряд дополнительных инструментов работы с результатами экспериментов. Для упорядочивания результатов, выбрав строку, есть возможность ее переместить вверх или вниз, нажав соответствующую кнопку. Выделенные строки с результатами можно удалить. Если выделить строку в таблице, а затем нажать на кнопку «Выделить строки с:», результаты экспериментов, у которых совпадают решаемая задача, метод и технология, будут выполнены для проведения одновременного сравнения.

5. Визуальное сравнение времени выполнения экспериментов. Визуальное сравнение численных данных позволяет быстрее сделать выводы относительно проведенных экспериментов. С этой целью внизу диалогового окна (рис. 4 область 4) строится гистограмма сравнения результатов вычислительных экспериментов. На гистограмме по горизонтальной оси отложен номер выполненного эксперимента, а по вертикальной время работы или ускорение. На графике указывается максимальное и минимальное время (ускорение), а также отношение максимального времени к минимальному. Рядом с каждым столбиком в гистограмме указывается время или ускорение.

Для обеспечения рассмотренных возможностей архитектура построения библиотеки параллельных алгоритмов и учебно-исследовательской системы ParaLib построена как система функциональной-ориентированных программных слоев - уровней (см. рис. 5).

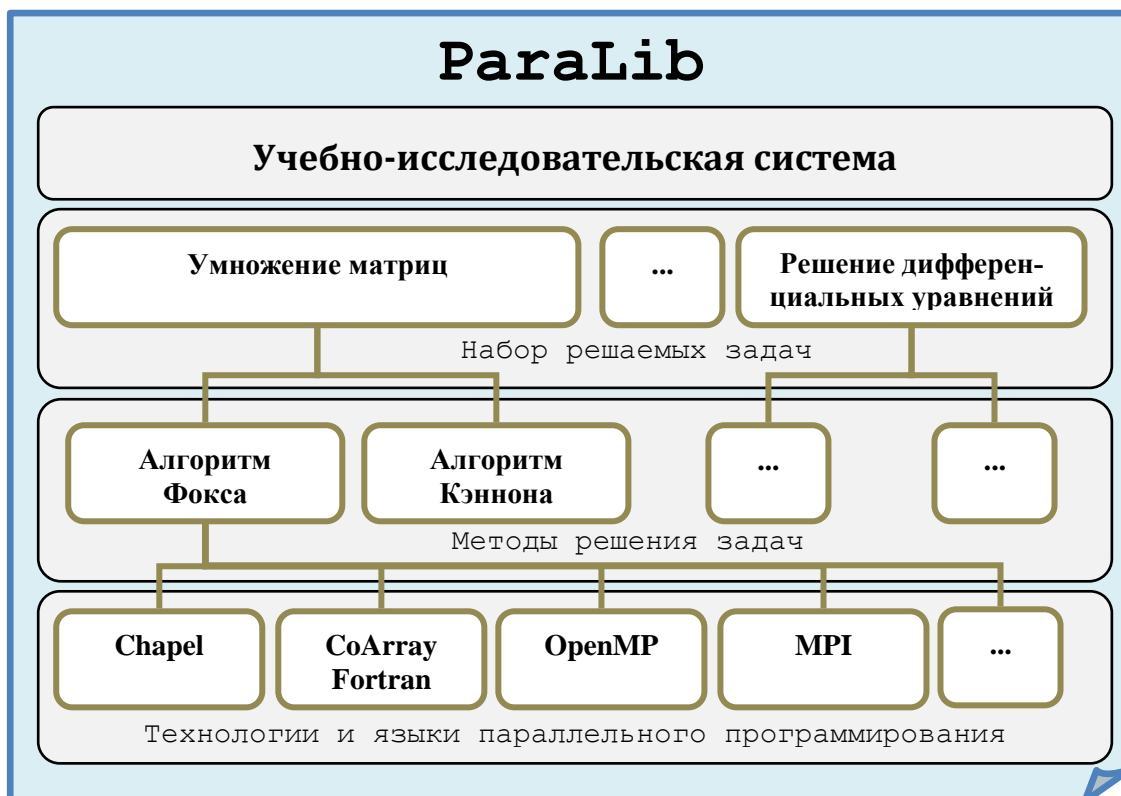


Рис. 5. Архитектура программного комплекса параллельных численных методов ParaLib

На верхнем уровне расположена учебно-исследовательская система. Система обеспечивает задание условий вычислительного эксперимента, организует выполнение исполняемых модулей при разных начальных условиях, позволяет накапливать информацию о проведенных экспериментах и делать выводы о применимости в рамках задачи выбранной технологии или языка параллельного программирования.

На следующем уровне расположена подсистема, объединяющая в своем составе вычислительные задачи, для которых в библиотеке имеются реализованные методы решения. Подсистема содержит также набор разработанных материалов с описаниями постановок всех представленных в библиотеке задач.

Далее располагается подсистема, в которой для каждой задачи предыдущего уровня представлены имеющиеся методы решения и их реализации с использованием разных языков и технологий параллельного программирования. Кроме того, подсистема содержит описания методов реализации параллельных алгоритмов.

На самом нижнем уровне расположена подсистема, содержащая реализации алгоритмов с применением различных методов, технологий и языков разработки параллельных программ. Каждая реализация в подсистеме представлена в виде нескольких файлов (файл с исходным кодом и исполняемый модуль). В рамках ParaLib считается, что исполняемый модуль на вход должен принимать четыре параметра – размер задачи, количество повторных запусков алгоритма, количество исполнителей (количество процессов или потоков) и возможно дополнительный параметр назначение которого определяется задачей. Результат работы исполняемого модуля – время работы каждого запуска алгоритма, которое записывается в отдельной строке журнала экспериментов.

Библиотека параллельных методов и учебно-исследовательская система ParaLib свободна для использования, загрузочный комплект может быть получен на сайте Центра суперкомпьютерных технологий Нижегородского университета [8].

Заключение

Система ParaLab и библиотека ParaLib носят учебно-исследовательский характер и могут быть использованы в ходе обучения специалистов различным аспектам параллельного программирования. Имеющиеся в программных комплексах реализации алгоритмов могут использоваться в качестве образцов при разработке параллельных программ с использованием различных языков и технологий параллельного программирования.

Литература

1. Гергель В.П. Теория и практика параллельных вычислений. - М.: Интернет-университет информационных технологий – ИНТУИТ.РУ, 2007. – 424 с.
2. Гергель В.П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем. – М.: Издательство Московского университета, 2010. – 567с.
3. Гергель В.П. Современные языки и технологии параллельного программирования // М.: Издательство Московского университета, 2012. – 408с.
4. Гергель В.П., Баркалов К.А., Мееров И.Б., Сысоев А.В. и др. Параллельные вычисления. Технологии и численные методы. Учебное пособие в 4 томах. – Нижний Новгород: Изд-во Нижегородского государственного университета, 2013. – 1394с.
5. Учебный курс CS338. Многопроцессорные вычислительные системы и параллельное программирование. URL: http://www.hpcc.unn.ru/mskurs/cs338_ppr_index.htm (дата обращения 10.05.2015)
6. John Reid, JKR Associates, UK Coarrays in the next Fortran Standard. April 21, 2010
URL:<ftp://ftp.nag.co.uk/sc22wg5/N1801-N1850/N1824.pdf>
7. The Chapel Parallel Programming Language. URL: <http://chapel.cray.com/> (дата обращения 10.05.2015)
8. Библиотека параллельных алгоритмов ParaLib. URL: <http://hpc-education.unn.ru/ru/paralib> (дата обращения 10.05.2015)