

Курс "Программирование и оптимизация для архитектуры Intel Xeon Phi"*

С.И. Бастраков, А.В. Горшков, А.В. Линев, И.Б. Мееров, А.Н. Свистунов, А.В. Сысоев

Нижегородский государственный университет им. Н.И. Лобачевского

В работе обсуждается проблема обучения программированию для Intel Xeon Phi. Представлен практико-ориентированный подход к обучению, основанный на обширном лабораторном практикуме. Отличительной особенностью подхода является комбинация классических задач на освоение концепций и приемов параллельного программирования и реальных примеров, возникших при решении научных задач финансовой математики и вычислительной физики. В работе приведено описание разработанного курса «Программирование для Intel Xeon Phi», представлена информация об опыте его прочтения. Материалы курса на русском языке выложены в открытый доступ на сайте www.intuit.ru [1-2].

1. Введение

Проблема выработки эффективных методик обучения параллельному программированию стоит перед научно-образовательным сообществом на протяжении многих лет, если определение «многих» в принципе применимо к данной, достаточно молодой области компьютерных наук. Несмотря на существенные достижения, к числу которых можно, в частности, отнести разработку единых сводов знаний [3-5], проблема полностью не решена. Сложность обучения различным аспектам параллельных вычислений заключена в самой природе параллелизма. Привыкшие к стройному последовательному потоку исполнения команд программисты нередко испытывают значительные затруднения при освоении новых непривычных моделей параллелизма. На первый взгляд может возникнуть впечатление, что освоить параллельное программирование просто. Изучение нескольких функций (MPI), прагм (OpenMP), автовекторизации циклов (SIMD), казалось бы, должно неизбежно привести к успеху. В ряде приложений это действительно так и есть, расчеты легко распараллеливаются естественным образом. Тем не менее, известно немало задач, эффективное распараллеливание в которых требует разработки новых алгоритмов, внедрения разных подходов к организации параллелизма (зачастую методом проб и ошибок). Подготовка специалистов, способных к такой работе, является весьма непростым делом.

Одним из важных факторов, с одной стороны определяющих мощь и выразительную силу параллельных технологий, но с другой стороны требующих постоянного совершенствования методик обучения, теоретических знаний и практических навыков, является развитие вычислительных архитектур. Одно из направлений развития связано с созданием высокопроизводительных ускорителей вычислений, примерами которых являются графические процессоры (GPU) и сопроцессоры Intel Xeon Phi. Кратко характеризуя состояние вопроса об обучении GPU-программированию, необходимо отметить, что несмотря на большое количество обучающей литературы [6-13], научиться эффективно программировать GPU весьма непросто. Не претендуя на истину в последней инстанции, отметим, что наша практика показывает, что системные программисты в основном осваивают GPU, тогда как прикладные программисты сталкиваются со значительными трудностями, в частности, из-за необходимости существенных изменений в своих программах. Для преодоления указанной проблемы при разработке Xeon Phi была заложена возможность после перекомпиляции выполнять традиционные x86-программы с достаточной степенью эффективности при условии их полного параллелизма, как на потоках, так и на уровне SIMD-инструкций. Но много ли таких приложений, где все и сразу получится само собой? Практика показывает, что эффективное использование Xeon Phi возможно не все-

* Авторы благодарят компанию Intel за финансовую поддержку разработки курса.

гда, а там, где возможно, нередко требует существенных знаний и навыков. Обсуждению вопроса об обучении программирования для Xeon Phi и посвящена данная работа.

В работе приводится описание подхода к обучению программированию на XeonPhi, разработанному в ННГУ при финансовой поддержке компании Intel. Отличительной особенностью указанного подхода является практико-ориентированное преподавание, сочетающее как классические примеры на организацию параллельных вычислений (матричные операции, сортировки и другие), так и авторские разработки, выросшие из реальных научных проектов из разных предметных областей (финансовая математика, вычислительная физика), выполняемых в университете. Работа построена следующим образом. В разделе 2 рассмотрены некоторые аналогичные работы. В разделе 3 дана краткая характеристика методики преподавания. В разделе 4 приведен учебно-тематический план лекционной части курса «Программирование для Intel Xeon Phi». В разделе 5 приведена информация об организации лабораторного практикума. В разделе 6 дана краткая характеристика методики оценки знаний обучаемых. В разделе 7 представлен опыт чтения курса на факультете ВМК ННГУ. В заключении приведено описание основных результатов и планов по их дальнейшему развитию.

2. Существующие курсы и материалы по программированию для Intel Xeon Phi

Область высокопроизводительных вычислений развивается очень интенсивно, практически каждый год происходит обновление актуальной совокупности аппаратных средств, технологий программирования и инструментальных средств, которыми должен владеть разработчик НРС-приложений. Подготовка высококвалифицированных специалистов в данной области требует формирования комплексных учебных программ, охватывающих различные аспекты обучения, и в то же время позволяющих оперативно их дополнять при появлении новых технологий. При появлении архитектуры Xeon Phi различные вузы использовали следующие основные подходы.

– Включение информации об архитектуре Xeon Phi и особенностях программирования для него в существующие курсы по архитектуре ЭВМ, параллельному программированию или смежным с ними [14-17].

Структура и содержание подобных курсов достаточно стандартны, поскольку широко обсуждались академическим сообществом и описаны в ряде разработанных рекомендаций (ACM Computer Science Curricula 2013 [3], NSF/IEEE-TCPP Curriculum Initiative [4], Свод знаний и умений в области СКТ [5] и т.п.) Архитектура Intel Xeon Phi использует элементы архитектуры Intel Pentium (P54C) и проекта Larrabee, и его изучение в рамках указанных курсов является естественным и обычно включает обзор аппаратных возможностей с обсуждением специфики, рассмотрение моделей программирования и обсуждение примеров решения задач.

– Привлечение сторонних организаций для прочтения курса.

Существует ряд организаций, разработавших курс или комплект курсов по программированию для Intel Xeon Phi и выполняющих обучение по данной теме по запросу [18-19]. В первую очередь, этим занимается сама компания Intel, подготовившая большой комплект учебных материалов и организовывающая обучение на многих мероприятиях, связанных с высокопроизводительными вычислениями [20], либо своими силами, либо с привлечением преподавателей из университетов. Обычно такие курсы читаются в форме мастер-классов с практическими занятиями, позволяющей за короткое время изучить особенности архитектуры и моделей программирования и попробовать их на практике.

– Разработка отдельного курса по программированию для Xeon Phi [1-2, 21-26].

В большинстве случаев подобный курс разрабатывается подразделением, занимающимся высокопроизводительными вычислениями, и проводится в форме мастер-классов и практических занятий. Но существуют и курсы классического типа (лекции + практика/лабораторные работы), в которых основное внимание уделяется практической части.

На наш взгляд все перечисленные подходы являются приемлемыми, и выбор зависит от структуры программы обучения в целом и целей обучения.

Основными источниками информации, использованными нами при разработке курса "Программирование и оптимизация для архитектуры Intel Xeon Phi" являлись:

- сайт компании Intel (разработчик архитектуры), содержащий техническую документацию, учебные материалы и статьи разработчиков [27-33];
- редкие статьи и книги, посвященные программированию для Intel Xeon Phi [34-36];
- наш собственный опыт разработки и переноса приложений в вычислительной физике и финансовой математике для Intel Xeon Phi.

Последняя из книг – "High Performance Parallelism Pearls: Multicore and Many-core Programming Approaches" [36] – включает главу 19 "Performance optimization of black-scholes pricing", подготовленную с участием двоих разработчиков представляемого курса.

3. Методика преподавания

Курс был разработан в нескольких вариантах - на русском и английском языках, в виде единого курса и разбитого на базовую и расширенную части, в виде модулей для представления на мероприятиях и мастер-классах. Далее мы будем рассматривать наиболее полный вариант курса.

Основная цель курса – изучить базовые принципы и сформировать навыки разработки программ, ориентированных на эффективное использование Intel Xeon Phi. При этом решаются следующие задачи:

- Изучение особенностей архитектуры Intel Xeon Phi, а также основных механизмов, влияющих на производительность программ.
- Изучение моделей использования Intel Xeon Phi, а также соответствующего системного программного обеспечения. Освоение способов разработки, сборки и запуска приложений на Intel Xeon Phi.
- Изучение принципов и особенностей применения технологий параллельного программирования для разработки и оптимизации расчетных программ, ориентированных на Intel Xeon Phi, включая вопросы использования инструкций SIMD, технологий OpenMP и Cilk Plus.
- Формирование навыков оптимизации и векторизации расчетных циклов, оптимизации работы с памятью, балансировки нагрузки при распараллеливании.
- Ознакомление с достаточно успешными примерами оптимизации программ, изначально не совсем подходящих для эффективного использования возможностей Intel Xeon Phi.

Курс рассчитан на 32 часа, в том числе:

- 10 часов лекций, на которых рассматриваются архитектура процессора, модели организации вычислений и подходы к оптимизации программ (см. п.4.);
- 10 часов практических занятий, которые могут проводиться в режиме лабораторных работ (слушатели пошагово выполняют задания под руководством инструктора или самостоятельно решают поставленную задачу, при необходимости пользуясь советами и подсказками преподавателя) либо в режиме мастер-классов (инструктор пошагово показывает то, что необходимо сделать, давая необходимые пояснения); темы лабораторных работ приведены в п.5;
- 12 часов выделяется на самостоятельную работу, в ходе которой студент должен выполнить практические домашние задания, а также выпускную работу, связанную с оптимизацией некоторого алгоритма для Intel Xeon Phi; примером задания для выпускной работы является достижение 50% от максимальной производительности в задаче умножения плотных матриц.

Для оценки знаний студентов используется система тестирования (см.п.6), и просмотр выполненных ими домашних заданий и выпускной работы.

Все материалы курса были разработаны на русском и английском языках. Русскоязычная версия курса представлена на сайте Интуит в виде двух связанных курсов для базового и продвинутого уровня [1-2].

4. Лекционная часть курса

Теоретическая часть включает следующие лекции.

1. Архитектура Intel Xeon Phi.
2. Выполнение программ на Intel Xeon Phi. Модели организации вычислений с использованием Intel Xeon Phi.
3. Векторные расширения Intel Xeon Phi.

4. Элементы оптимизации прикладных программ для Intel Xeon Phi. Intel C/C++ Compiler.

5. Элементы оптимизации прикладных программ для Intel Xeon Phi: Intel MKL, Intel VTune Amplifier XE.

Фактически, лекционная часть курса делится на три блока: знакомство с архитектурой Intel Xeon Phi, изучение моделей программирования, обеспечение эффективного использования вычислительной мощности архитектуры (если задача это допускает).

В лекции 1 рассматриваются основные архитектурные блоки и особенности сопроцессора: ядро, блок векторной обработки данных, встроенная высокопроизводительная двунаправленная кольцевая шина, полностью когерентные кэши L2 и принципы взаимодействия компонент. Описывается принцип работы конвейера ядра, рассматривается иерархия памяти. Основное внимание уделяется элементам, наиболее существенно влияющим на производительность вычислений, и пониманию архитектурной основы способов оптимизации программ для архитектуры Intel Xeon Phi.

Лекция 2 содержит описание принципов выполнения программ и моделей организации вычислений с использованием сопроцессора Intel Xeon Phi (offload model, coprocessor-only model, symmetric model). Обсуждаются способы создания приложений для Xeon Phi. Описываются архитектура и состав ПО, обеспечивающего выполнение программ на Intel Xeon Phi. Дается краткое описание Intel Manycore Platform Software Stack и симметричного коммуникационного интерфейса.

В начале лекции 3 дается вводная информация о векторных расширениях. Далее рассматриваются особенности векторных расширений Intel Xeon Phi: дается краткое описание основных типов данных и регистрового пула, приводится обзор основных типов операций, упоминается расширенная поддержка математических функций. Рассматривается вопрос о доступе к векторным расширениям при программировании на языках высокого уровня. Описываются несколько способов векторизации расчетов и их особенности. В заключительной части лекции затрагивается важный вопрос о связи векторизации с вычислением математических функций.

Лекция 4 посвящена изучению некоторых приемов оптимизации кода для Intel Xeon Phi. В первой части лекции обсуждаются новые элементы языков программирования, предназначенные для работы с памятью при использовании режима offload. Рассматриваются явная и неявная схемы работы с памятью, дается их сравнительная характеристика. Во второй части приводится дополнительная информация о способах векторизации кода, не рассмотренная подробно в предыдущей лекции. Вновь обсуждается автоматическая векторизация кода, использование директив для векторизации, новые возможности Array notation и Elemental functions, а также другие возможности компилятора. В третьей части рассматривается встроенный в компилятор профилировщик циклов, обсуждается использование отчетов компилятора, балансировка нагрузки в параллельных программах, даются некоторые дополнительные рекомендации. Изложение сопровождается примерами программ.

В лекции 5 изучаются два важных и активно применяемых инструмента из пакета Intel Parallel Studio XE – профилировщик Intel VTune Amplifier XE (далее Amplifier) и библиотека математических алгоритмов Intel Math Kernel Library (MKL) – и возможности и правила их использования при программировании для Intel Xeon Phi. Вначале изучаются особенности использования библиотеки MKL при программировании на сопроцессоре: возможности отправки части вычислений на сопроцессор в режимах Automatic offload, Compiler assisted offload, выполнение расчетов на Xeon Phi и рекомендации по выбору модели программирования. Затем рассматривается оптимизация приложений с помощью Amplifier: приводится обзор инструмента, рассказывается об анализе эффективности приложений на сопроцессоре, обсуждаются некоторые важные метрики для оценки производительности.

5. Лабораторный практикум

Комплект лабораторных работ обеспечивает получение базовых навыков работы с системами, использующими Intel Xeon Phi, включает рассмотрение стандартных задач математического программирования, а также знакомство с ситуациями, возникающими при решении реальных научно-исследовательских задач в области вычислительной физики и финансовой математики.

1. Компиляция и запуск приложений на Intel Xeon Phi.
2. Оптимизация прикладных программ для Intel Xeon Phi с использованием Intel C/C++ Compiler. Векторизация.
3. Оптимизация вычислений в задаче о разложении чисел на простые множители. Векторизация и балансировка нагрузки.
4. Оптимизация вычислений в задаче о вычислении справедливой цены опциона Европейского типа. Оптимизация: шаг за шагом.
5. Оптимизация вычислений в задаче матричного умножения. Оптимизация работы с памятью.

В лабораторной работе 1 (ЛР1) изучаются основные модели программирования и режимы компиляции и запуска программ, а также рассматривается ряд примеров, демонстрирующих процесс переноса кода на сопроцессор Intel Xeon Phi. Демонстрируются подходы к написанию кода в каждом из режимов и последовательность шагов, которые необходимо выполнить для компиляции и запуска программ. В рамках данной лабораторной работы рассматриваются режимы offload, coprocessor only и symmetric mode.

ЛР2 формирует навыки векторизации кода в программах на языке C с использованием Intel C/C++ Compiler. Иллюстрируются основные способы использования векторных инструкций на сопроцессоре Intel Xeon Phi: использование ключевого слова restrict и #pragma ivdep для гарантии отсутствия потенциальных зависимостей, явная векторизация через #pragma simd, использование Array notation и Elemental functions, рассматриваются вопросы векторизации циклов с вызовами математических функций. В завершение рассматриваются более сложные примеры. Акцент в ЛР2 сделан на обзорном рассмотрении различных средств на простых примерах и представлении многообразия доступных в настоящее время средств векторизации и общей идеологии их использования.

В ЛР3 иллюстрируются некоторые решения, позволяющие добиться результата там, где этого, казалось бы, сделать не удастся. Рассмотрен один из самых простых алгоритмов решения задачи разложения натуральных чисел на простые множители (выбор алгоритма обусловлен не его эффективностью, а удобством для демонстрации приемов оптимизации.) Описывается последовательная реализация алгоритма, выполняется перенос вычислений на сопроцессор. Предлагается простейшая параллельная реализация алгоритма, выполняется анализ ее эффективности, обсуждаются возможные способы балансировки нагрузки для повышения производительности. Демонстрируется один из методов векторизации сложных циклов. Показывается способ построения гибридной схемы вычислений, позволяющей задействовать в расчетах CPU и Xeon Phi.

В ЛР4 иллюстрируются некоторые приемы оптимизации вычислений в научных и инженерных приложениях, рассматривается достаточно простая задача финансовой математики – вычисление справедливой цены опциона Европейского типа в том случае, когда эта цена может быть вычислена аналитически. Дается краткое описание модели финансового рынка, формулируются основные понятия, приводится их интуитивное описание, способствующее пониманию сути и смысла решаемой прикладной задачи. Далее описывается базовая программная реализация, проводится анализ производительности, шаг за шагом в программу вносятся изменения, иллюстрирующие приемы оптимизации: устранение ненужных преобразований типов, вынос инвариантов, эквивалентные преобразования с заменой «тяжелых» математических функций более «легкими», векторизация вычислений, распараллеливание, «прогрев» – снижение накладных расходов на создание потоков, понижение точности вычислений с плавающей запятой, оптимизация работы с памятью (использование streaming stores). Эффекты от оптимизации демонстрируются как на Intel Xeon, так и на Intel Xeon Phi.

ЛР5 посвящена изучению базовых принципов применения известных способов повышения производительности на примере известной, широко распространенной в приложениях и хорошо исследованной в литературе задачи – умножения плотных матриц. В отличие от предыдущих лабораторных работ, подробно изучающих разные техники оптимизации расчетов, основное внимание уделяется вопросам оптимизации работы с памятью. Для сравнения результатов вычислений и времени работы реализуется умножение плотных матриц с использованием высокопроизводительной библиотеки Intel MKL. Приводится наивная реализация алгоритма умножения матриц. Рассматриваются вопросы влияния порядка циклов на время вычислений на

сопроцессор Intel Xeon Phi. Изучается влияние выравнивания и подсказок компилятору на векторизацию вычислений. Реализуется несколько редакций блочного алгоритма умножения матриц. Формулируются задания для самостоятельной проработки.

При выполнении лабораторных работ и проведении мастер-классов мы используем компьютер "Лобачевский" Нижегородского государственного университета, который включает несколько узлов, оснащенных сопроцессорами Intel Xeon Phi [37].

6. Методика оценки знаний обучаемых

Оценка знаний студентов производится на основании следующих данных.

- Оценки выполнения домашних заданий.
- Результат прохождения теста по материалам курса.

В ходе разработки курса было подготовлено более 200 вопросов, из которых составлены тесты по курсу. Тесты представлены на сайте Интуит в составе курса (www.intuit.ru) и на сайте программы сертификации ННГУ - Intel (nncc.unn.ru).

- Результат выполнения выпускной работы.

В выпускной работе оценивается достигнутый уровень производительности и тип и качество проведенных оптимизаций.

7. Опыт прочтения курса на факультете ВМК ННГУ

Занятия по курсу «Программирование на Intel Xeon Phi» на третьем курсе факультета ВМК ННГУ проводились по следующей схеме. В начале читалось несколько обзорных лекций, включая «Введение в параллельное программирование на современных архитектурах», «Микроархитектура Intel Xeon Phi» и «Модели исполнения программ на сопроцессоре». Затем каждая лекция чередовалась с практическим занятием, на котором студентам предлагалось самостоятельно выполнить 2-3 небольших задания по материалу, рассказанному на предыдущей лекции. Примерами таких пар «лекция-практика» могут служить занятия по темам «Программирование в режиме Offload», «Векторизация», «Подходы к оптимизации программ на сопроцессоре» и др. Практические занятия проводились на базе вычислительного кластера ННГУ "Лобачевский" с Intel Xeon Phi на узлах. Каждый студент имел доступ к кластеру со своей локальной машины по протоколу SSH.

Что касается степени усваивания материалы студентами, то по нашим наблюдениям лучше всего усваиваются те части программы, акцент на которые делается в практической части курса. В то же время материал лекций, не вполне подкрепленных практикой, усваивается хуже. Это справедливо, например, к теме «Микроархитектура Intel Xeon Phi». По окончании курса у большинства студентов остается только базовое понимание архитектуры сопроцессора, в то время как низкоуровневые детали остаются без внимания.

Отметим также, что данный курс опирается на уже изученные студентами технологии параллельного программирования, такие как OpenMP и MPI, повторное описание которых не входило в материал. Однако проблем с их применением на практике в рамках новой инфраструктуры замечено не было.

Основной технической сложностью при проведении занятий является необходимость взаимодействия студентов с кластером для доступа непосредственно к Intel Xeon Phi. Причем проблем здесь две.

Первая состоит в том, что в стандартную учебную программу не входит обучение студентов работе с системами управления кластером, а значит, это приходится делать в рамках данного курса. Причем в силу ограниченности времени студентам дается не общее описание систем управления, а скорее набор правил, команд, которые нужно выполнить для запуска задачи на сопроцессоре. Это приводит к тому, что любое отклонение от стандартных команд оборачивается проблемами с запуском, которые приходится решать индивидуально с каждым студентом во время практических занятий.

Вторая проблема – наличие свободных ресурсов кластера. В рамках апробации курса проблему свободных ресурсов мы решали путем выделения отдельного раздела, состоящего из не-

скольких узлов кластера, доступ к которым могли иметь только студенты во время практических занятий.

В ходе апробации техника проведения зачета по курсу сводилась к следующему. Степень выполнения задач, предлагаемых студентам в рамках каждого из практических занятий, фиксировалась преподавателем. По результатам семестра студенты, которые успешно справились с большей частью заданий, получали зачет автоматом. Следует отметить, что в рамках апробации таких студентов было большинство. С остальными студентами проводились устные беседы на знание основ программирования на Intel Xeon Phi.

По окончании данного курса студенты получили как теоретические знания об устройстве сопроцессора, методах его программирования и основных подходах к оптимизации программ для Intel Xeon Phi, так и практические навыки программирования сопроцессора в различных режимах.

8. Заключение

Мы представили учебный курс "Программирование и оптимизация для архитектуры Intel Xeon Phi", разработанный в Нижегородском государственном университете им.Н.И. Лобачевского. Основной целью курса является изучение базовых принципов написания высокопроизводительных приложений для архитектуры Intel Xeon Phi. Мы представили программу курса и методику преподавания, которые, по нашему мнению, позволяют подготовить специалистов, способных принимать участие в научных исследованиях в области вычислительной физики и финансовой математики. По результатам проделанной работы планируется издать учебное пособие.

Литература

1. Введение в принципы функционирования и применения современных мультиядерных архитектур (на примере Intel Xeon Phi)
URL: <http://www.intuit.ru/studies/courses/10611/1095/info> (дата обращения: 14.06.2015).
2. Программирование на современных мультиядерных архитектурах (на примере Intel Xeon Phi)
URL: <http://www.intuit.ru/studies/courses/10612/1096/info> (дата обращения: 14.06.2015).
3. Computer Science Curricula 2013
URL: <https://www.acm.org/education/CS2013-final-report.pdf> (дата обращения: 14.06.2015).
4. NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing - Core Topics for Undergraduates
URL: <http://www.cs.gsu.edu/~tcpp/curriculum/sites/default/files/NSF-TCPP-curriculum-version1.pdf> (дата обращения: 14.06.2015).
5. Свод знаний и умений предметной области «Суперкомпьютеры и параллельные вычисления»
URL: http://hpc-education.ru/files/Svod_v24_intro_2013_09_20.pdf (дата обращения: 14.06.2015).
6. Cheng J., Grossman M., McKercher T. Professional CUDA C Programming. N.-Y.: Wrox .2014. - 528p.
7. Cook Sh. CUDA Programming: A Developer's Guide to Parallel Computing with GPUs. Morgan Kaufmann, 2012. — 600 p.
8. Farber R. CUDA Application Design and Development. Morgan Kaufmann, 2011. — 336 p.
9. Fernando R. (ed.). GPU Gems: Programming techniques, tips and tricks for real-time graphics. Addison-Wesley, 2004. — 784 p.

10. Hwu W.W. GPU Computing Gems Jade Edition. Applications of GPU Computing Series. — Morgan Kaufmann, 2011. — 560 p.
11. Hwu W.W. GPU Computing Gems: Emerald Edition. Morgan Kaufmann, 2011. - 886 p.
12. Sanders Jason, Kandrot Edward. CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley Professional, 2010. — 312 p.
13. Боресков А.В. и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учеб. пособие / А. В. Боресков и др. Предисл.: В. А. Садовничий. - М.: Издательство Московского университета, 2012. - 336 с.
14. 6.888 Parallel and Heterogeneous Computer Architecture
URL: <http://courses.csail.mit.edu/6.888/spring13/syllabus.shtml> (дата обращения: 14.06.2015).
15. MCS 572: Introduction to Supercomputing
URL: <http://homepages.math.uic.edu/~jan/mcs572/> (дата обращения: 14.06.2015).
16. CIS*3090 Parallel Programming
URL: <http://www.uoguelph.ca/~gardnerw/courses/cis3090/> (дата обращения: 14.06.2015).
17. Advanced Computational Physics - Parallel Computing
URL: <http://einstein.drexel.edu/courses/PHYS405/Content.html> (дата обращения: 14.06.2015).
18. Crash Course on Programming and Optimization with Intel Xeon Phi Coprocessors at SC14
URL: <http://research.colfaxinternational.com/post/2014/11/16/SC14.aspx> (дата обращения: 14.06.2015).
19. Parallel Programming and Optimization for Intel® Xeon Phi™ Coprocessors
URL: <http://www.accelware.com/xeon-phi-training> (дата обращения: 14.06.2015).
20. Учебные курсы по возможностям сопроцессора Intel Xeon Phi | Intel Developer Zone
URL: <https://software.intel.com/ru-ru/mic-developer/training> (дата обращения: 14.06.2015).
21. Master Class: Programming on Supercomputers.
URL: <https://www.itmasters.edu.au/free-short-course-programming-super-computers/> (дата обращения: 14.06.2015).
22. Parallel Programming and Optimization With Intel Xeon Phi Coprocessor.
URL: <https://www.msi.umn.edu/content/parallel-programming-and-optimization-intel-xeon-phi-coprocessor> (дата обращения: 14.06.2015).
23. Introduction to Programming the Xeon Phi Processor
URL: <https://www.pdc.kth.se/events/event-repository/introduction-to-programming-the-xeon-phi-processor-november-2015> (дата обращения: 14.06.2015).
24. Programming the Xeon Phi (XSEDE)
URL: <https://portal.tacc.utexas.edu/-/programming-the-xeon-phi-xsede-> (дата обращения: 14.06.2015).
25. Курс "Дополнительные главы по гетерогенным параллельным вычислениям"
URL: <http://msu-intel.rubea.net/node/103> (дата обращения: 14.06.2015).
26. Дисциплина по выбору "Программирование Intel Xeon Phi"
URL: <http://ssd.sccc.ru/ru/chair/nsu/phi> (дата обращения: 14.06.2015).
27. Intel Developer Zone
URL: <http://software.intel.com/en-us/mic-developer> (дата обращения: 14.06.2015).
28. Intel Xeon Phi Coprocessor System Software Developers Guide
URL: <http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-system-software-developers-guide> (дата обращения: 14.06.2015).

29. Nguyen Loc Q et al. Intel Xeon Phi Coprocessor Developer's Quick Start Guide. URL: <http://software.intel.com/en-us/articles/intel-xeon-phi-coprocessor-developers-quick-start-guide> (дата обращения: 14.06.2015).
30. Rahman R. Intel Xeon Phi Core Micro-architecture URL: <http://software.intel.com/en-us/articles/intel-xeon-phi-core-micro-architecture> (дата обращения: 14.06.2015).
31. Krishnai R. Data Alignment to Assist Vectorization URL: <http://software.intel.com/en-us/articles/data-alignment-to-assist-vectorization> (дата обращения: 14.06.2015).
32. Mackay D. Optimization and Performance Tuning for Intel Xeon Phi Coprocessors, Part 1: Optimization Essentials, 2012. URL: <http://software.intel.com/en-us/articles/optimization-and-performance-tuning-for-intel-xeon-phi-coprocessors-part-1-optimization> (дата обращения: 14.06.2015).
33. Reinders J. An Overview of Programming for Intel Xeon processors and Intel Xeon Phi coprocessors. URL: <http://software.intel.com/en-us/blogs/2012/11/14/an-overview-of-programming-for-intel-xeon-processors-and-intel-xeon-phi> (дата обращения: 14.06.2015).
34. Jeffers J., Reinders J. Intel Xeon Phi Coprocessor High-Performance Programming. Morgan Kaufmann Publishing and Elsevier, 2013.
35. Jeffers J., Reinders J. High Performance Parallelism Pearls: Multicore and Many-core Programming Approaches. Morgan Kaufmann, 2014.
36. Vladimirov A., Asai R., Karpusenko V. Parallel Programming and Optimization with Intel Xeon Phi Coprocessors. Colfax International, 2015.
37. Приволжский научно-образовательный центр суперкомпьютерных технологий - Ресурсы URL: <http://hpc-education.unn.ru/ru/Ресурсы> (дата обращения: 14.06.2015).