

Обучение основам разработки приложений для современных архитектур высокопроизводительных вычислительных систем*

С.А. Немнюгин

Санкт-Петербургский государственный университет (СПбГУ), Санкт-Петербург

В данной работе обсуждаются структура и наполнение учебного курса, посвященного методам создания прикладного программного обеспечения для современных и перспективных компьютерных архитектур. Основное внимание уделяется разработке гибридных приложений.

Ключевые слова: обучение, гибридные архитектуры, программное обеспечение, алгоритмы.

1. Введение

Проект создания курса «Разработка гибридного прикладного программного обеспечения» направлен на разработку учебного курса для магистратуры, посвященного методам создания прикладного программного обеспечения для современных и перспективных компьютерных архитектур. Такие архитектуры вычислительных систем как высокопроизводительных, так и класса автоматизированных рабочих мест исследователя и технического специалиста нередко являются гибридными, то есть включают в свой состав вычислительные устройства разного типа. Эффективное программирование для этих устройств основано на разных подходах, использовании разного программного инструментария и разных алгоритмов. Целью данного проекта является разработка курса, который позволит дать специалисту-разработчику прикладного программного обеспечения необходимые компетенции в области эффективного использования гибридных вычислительных систем.

Технологии высокопроизводительных вычислений позволяют решать сложные научные, технические и другие задачи, представляющие значительный интерес не только с академической, но и с практической точки зрения. Решение ряда таких задач имеет особую социальную значимость, так как связано с развитием новых технологий и созданием нового качества жизни, повышением безопасности технических конструкций и транспортных средств, развитием медицины, уменьшением зависимости человека от погодных и климатических факторов и т.д. Реалистические модели, используемые в вышеперечисленных областях, довольно сложны и могут быть исследованы только численно, с помощью высокопроизводительных систем. Доминирующим типом архитектур этих систем являются и будут оставаться в течение достаточно длительного времени гибридные архитектуры, включающие в свой состав вычислительные устройства разного типа: серверные процессоры, ускорители вычислений типа «графический процессор общего назначения», сопроцессоры Intel® Xeon Phi и другие. Для каждого из этих типов вычислителей имеются свои подходы к программированию, свои методы оптимизации, свой программный инструментарий. Так, например, для реализации параллельных вычислительных алгоритмов на «обычных» многоядерных процессорах используется OpenMP, а для графических процессоров

* Автор выражает признательность Научному парку СПбГУ и Ресурсному образовательному центру «Физика» за поддержку данного проекта.

общего назначения - OpenCL, CUDA или другие подходы. Если вычислители разного типа объединяются в одной гибридной архитектуре, требуется учет особенностей этих архитектур и «гибридизация» разработки прикладного программного обеспечения. Наличие соответствующих навыков и компетенций у специалиста становится, таким образом, важнейшей составной частью его профессиональной культуры.

2. Структура, наполнение курса и его целевая аудитория

В настоящее время курсы по технологиям параллельного программирования и различным аспектам высокопроизводительных вычислений включены в магистерские и бакалаврские программы ведущих университетов Российской Федерации. Это Московский государственный университет им. М.В.Ломоносова, Санкт-Петербургский государственный университет, Нижегородский государственный университет им. Н.И.Лобачевского, Южно-Уральский государственный университет и ряд других. Общим для большинства курсов является то, что каждый такой курс обычно посвящен одной определенной технологии программирования высокопроизводительных вычислений (например, «Программирование на OpenMP», «Программирование на MPI») или одной теме, рассматриваемой достаточно изолированно (например, «Параллельные алгоритмы»). Вышеупомянутыми особенностями обладают также дистанционные курсы. Наиболее известным в Российской Федерации проектом такого рода является «Интернет-университет суперкомпьютерных технологий», возникший в рамках проекта ИНТУИТ. Вместе с тем очевидна необходимость курсов, представляющих комплексный, интегрированный подход и ориентированных на гибридные архитектуры вычислительных систем.

Предлагаемый к разработке курс состоит из следующих разделов:

1. Современные архитектуры вычислительных систем вообще и высокопроизводительных систем в частности, тенденции их развития. Современный серверный процессор. Графические процессоры общего назначения. Ускорители вычислений Intel® Xeon Phi. Другие типы сопроцессоров. Анализ рейтинга Top500 Supercomputers. Обзор архитектурных особенностей мобильных и встраиваемых систем. Гибридные архитектуры первого и второго типа.
2. Принципы параллельного программирования, теоретический анализ эффективности и масштабируемости высокопроизводительных вычислений, теоретические модели параллельных вычислений и законы. Модель вычислений и законы Амдала. Законы Густафсона-Барсиса и Сана-Ная. Особенности программирования для мобильных и встраиваемых систем.
3. Факторы, влияющие на эффективность приложений. Мощность арифметических и логических выражений. Кэш-эффективность. Расположение данных. Предсказание ветвлений и другие аспекты. Необходимость контроля корректности программного обеспечения при его оптимизации.
4. Многопоточный параллелизм и его реализация на серверных процессорах. Особенности, проблемы эффективной реализации, условия их появления и способы разрешения. Краткий обзор технологий программирования, в том числе OpenMP, Cilk. Обзор алгоритмов, которые эффективно отображаются на данную архитектуру.
5. Многопоточный параллелизм и его реализация на ускорителях вычислений. Учет особенностей архитектуры ускорителей, специфические проблемы эффективной реализации, условия их появления и способы разрешения. Концепция выгрузки вычислений. Краткий обзор технологий программирования. Обзор алгоритмов, которые эффективно отображаются на данную архитектуру.
6. Многопоточный параллелизм на гибридных архитектурах. Совместное использование разных технологий программирования. Декомпозиция алгоритма на часть, предназначенную для реализации на хост-процессоре и часть, реализуемую для сопроцессора.
7. Многозадачный параллелизм. Краткий обзор технологии программирования MPI (Message Passing Interface). Многозадачность и многопоточность в одном приложении. Обзор алгоритмов, которые эффективно отображаются на архитектуру кластерных вы-

числительных систем. Декомпозиция алгоритма на часть, которая эффективно реализуется в рамках многозадачного параллелизма, и части многопоточного параллелизма разного вида. MPI вместе с OpenMP, потокобезопасность.

Методология преподавания данного курса предполагает тесную интеграцию нескольких форм методической работы. Во-первых, это теоретические лекции. Во-вторых, это демонстрация теоретических положений с помощью заранее подготовленных относительно простых примеров программ. Третий компонент занятий в рамках данного курса - выполнение более сложных исследовательских заданий, содержанием которых является применение изучаемых подходов, исследование их эффективности, масштабируемости и иных свойств. Методическим сопровождением третьего компонента данного курса будут лабораторные работы, включающие описания, краткие инструкции преподавателю, шаблоны программ. В тесной интеграции вышеперечисленных трех компонентов состоит определенная методическая новизна, так как обычно курсы по информационным технологиям включают только лекции или лекции и семинары, посвященные решению задач стандартного типа. Методическая новизна представленного проекта заключается также и в его наполнении - предполагается рассматривать вопросы разработки эффективного программного обеспечения именно для гибридных архитектур.

Для реализации данного проекта необходим компьютерный класс, имеющий гибридную архитектуру и позволяющий продемонстрировать различные технологии программирования для таких систем. Требуется также программное обеспечение, предназначенное для разработки – интегрированные среды, компиляторы, средства динамического анализа приложений. В Ресурсном образовательном центре «Физика» Санкт-Петербургского государственного университета имеется компьютерный класс, отвечающий этим требованиям. Он состоит из 11 посадочных мест, объединенных локальной сетью в учебный вычислительный кластер. Каждое посадочное место (вычислительный узел кластера) содержит 1 или 2 процессора Intel® Xeon E5-2690 (количество физических ядер 8). На каждом узле имеются 1 или 2 ускорителя вычислений для выполнения гибридных вычислений: графический процессор общего назначения NVIDIA Tesla или Intel® Xeon™ Phi. Класс работает под управлением операционной системы Microsoft Windows Server. В классе установлены лицензионные средства профессиональной разработки программного обеспечения (ПО) от корпораций Microsoft и Intel. Имеются средства анализа и оптимизации ПО. Есть также средства программирования параллельных и высокопроизводительных вычислений. Методическое обеспечение основано на [1-7].

Данный проект является развитием существующих учебных программ и курсов основной образовательной программы "Прикладные физика и математика" по направлению "Прикладные математика и физика", уровень магистратура. Он соответствует образовательным стандартам Санкт-Петербургского государственного университета и направлен на формирование следующих профессиональных компетенций:

1. Знание и понимание особенностей архитектур современных вычислительных систем и тенденций их развития.
2. Знание и понимание того, как особенности архитектур гибридных вычислительных систем влияют на эффективность программного обеспечения.
3. Знание технологий программирования высокопроизводительных вычислений для серверных процессоров, а также разных типов ускорителей вычислений.
4. Знание возможных причин некорректной работы программного обеспечения, причин потери эффективности и методов преодоления этих проблем.
5. Знание теоретических законов масштабируемости программного обеспечения.
6. Умение совместно использовать различные технологии разработки прикладного программного обеспечения для высокопроизводительных вычислений.
7. Умение оптимизировать программное обеспечение.

Целевая аудитория курса – студенты магистратуры Санкт-Петербургского государственного университета. Эта категория обучающихся имеет достаточную базу знаний по информационным технологиям, программированию, математике и предметной области своей научной специализации.

Литература

1. С.А.Немнюгин, Введение в программирование на кластерах. НОУ ИНТУИТ, 2016, 247 с. (Электронная книга).
2. С.А.Немнюгин, Введение в программирование на Intel® Cilk Plus. НОУ ИНТУИТ, 2016, 147 с. (Электронная книга).
3. С.А.Немнюгин, Модели и средства программирования для многопроцессорных вычислительных систем. НОУ ИНТУИТ, 2016, 189 с. (Электронная книга).
4. С.А.Немнюгин, Программирование на Intel® Cilk Plus. НОУ ИНТУИТ, 2016, 193 с. (Электронная книга).
5. С.А.Немнюгин, Программирование на кластерах с использованием инструментов Intel (Intel® Cluster Studio). НОУ ИНТУИТ, 2016, 259 с. (Электронная книга).
6. Методические материалы Международной молодежной научно-практической школы «Высокопроизводительные вычисления на Grid системах». Адрес размещения в Интернете:
http://itprojects.narfu.ru/grid/materials2017/nemnyugin_grid_school_2017.pdf
7. Методические материалы Интернет-университета суперкомпьютерных технологий. Адрес размещения в Интернете:
<http://www.hpcu.ru/courses/39>

Teaching to basics of software development for modern architectures of high-performance computing systems

S.A.Nemnyugin

Saint-Petersburg State University (SPbSU), Saint-Petersburg

Abstract. Structure and content of the course devoted to methods of development of software for modern and perspective computer architectures is discussed. Main focus is on the development of hybrid applications.

Keywords: learning, hybrid architectures, software, algorithms

References

1. Nemnyugin S.A. Vvedenie v programmirovaniye dlya klasterov [Introduction in programming for clusters]. NOU INTUIT, 2016. 247 P.
2. Nemnyugin S.A. Vvedenie v programmirovaniye na Intel® Cilk™ Plus [Introduction in programming on Intel® Cilk™ Plus]. NOU INTUIT, 2016. 147 P.
3. Nemnyugin S.A. Modeli i sredstva programmirovaniya dlya mnogoprocessornykh vychislitelnykh sistem [Models and tools of programming for multiprocessor computing systems]. NOU INTUIT, 2016. 189 P.
4. Nemnyugin S.A. Programmirovaniye na Intel® Cilk™ Plus [Programming on Intel® Cilk™ Plus]. NOU INTUIT, 2016. 193 P.
5. Nemnyugin S.A. Programmirovaniye na klasterah s ispolzovaniem instrumentov Intel (Intel® Cluster Studio) [Programming on clusters with Intel software tools (Intel® Cluster Studio)]. NOU INTUIT, 2016. 259 P.
6. Nemnyugin S.A. Metodicheskie materialy Mezhdunarodnoi molodezhnoi nauchno-prakticheskoi shkoly “Vysokoproizvoditelnye vychisleniya na GRID-sistemakh” [Methodical materials of International Youth Scientific and Practical Workshop “High-Performance Computing on GRID-systems”]. Arkhangelsk 2017. Web-link: http://itprojects.narfu.ru/grid/materials2017/nemnyugin_grid_school_2017.pdf.
7. Nemnyugin S.A. Metodicheskie materialy Internet-universiteta superkomputernykh tekhnologij [Methodical materials of Internet University of Supercomputer Technologies]. Web-link: <http://www.hpcu.ru/courses/39>.