

# ОБ ОПЫТЕ ПРЕПОДАВАНИЯ ТЕХНОЛОГИЙ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛЕНИЙ

А.А. Захаров, И.Г. Захарова

В настоящее время можно выделить множество направлений подготовки бакалавров и магистров в общей номенклатуре образовательных программ системы высшего профессионального образования, для которых такая компетенция, как владение технологиями высокопроизводительных вычислений (НРС) просто необходима. Хотя эта задача была очевидна достаточно давно, своего отражения в содержании Федеральных государственных стандартов высшего профессионального образования (ФГОС ВПО) в полной мере она не нашла. Однако все прописать в стандартах невозможно, и многое подразумевается по умолчанию. В частности, указанная компетенция непосредственно связана, например, с такими профессиональными компетенциями, как «знать проблемы и направления развития технологий программирования»; «знать направления развития компьютеров с традиционной (нетрадиционной) архитектурой; тенденции развития функций и архитектур проблемно-ориентированных программных систем и комплексов» из ФГОС ВПО направления «Математическое обеспечение и администрирование информационных систем». Аналогичные или близкие по содержанию компетенции можно найти в стандартах и других ИТ-направлений.

Тюменский государственный университет (ТюмГУ) участвует в проекте «Национальная суперкомпьютерная технологическая платформа», который реализуется для решения вопросов развития высокопроизводительных вычислений в России [1]. ТюмГУ имеет хорошую материальную базу для обучения технологиям высокопроизводительных вычислений, которое осуществляется с 2001 г. При этом, практически в полном соответствии с общемировыми тенденциями того времени [2], подготовка изначально проводилась в рамках отдельного (обязательного для ГОС ВПО 2-ого поколения) курса («Параллельное программирование», 7 семестр) только на специальности «Математическое обеспечение и администрирование информационных систем» со специализацией «Технологии программирования». Акцент делался на практическое освоение технологии MPI на базе C/C++. В дальнейшем, по инициативе преподавателей отдельные разделы, связанные с НРС, были включены в курсы «Объектно-ориентированное программирование» (3-4 семестры, программирование потоков на стандартных классах), «Структуры и алгоритмы компьютерной обработки данных» (3 семестр, параллельные алгоритмы для задач поиска и сортировки), «Сетевые технологии» (5-6 семестры, модели передачи сообщений). Таким образом, постепенно была организована своего рода непрерывная подготовка по НРС. В какой-то мере все это помогло открыть профиль подготовки «Параллельное программирование» (направление «Математическое обеспечение и администрирование информационных систем»), в рамках которого в дополнение к вышеперечисленным дисциплинам предусмотрены пять специальных курсов, включающих изучение как теоретических, так и прикладных аспектов НРС.

Однако, то, что вполне реализуемо для целенаправленного обучения программистов, невозможно на других направлениях и специальностях – как в силу недостатка времени, отводимого на изучение соответствующих дисциплин, так и по причине других целей подготовки студентов. Открытыми остаются вопросы о том, обязательной или факультативной должна быть соответствующая дисциплина, читать ли ее на младших или на старших курсах, и, главное, - это должна быть одна дисциплина или все-таки несколько.

Для ответа на эти вопросы представляется необходимым выделить для всех ИТ-специальностей и направлений три довольно грубых условных уровня требований к подготовке в области НРС в соответствии с реалиями будущей работы: сопровождение, разработка, анализ кода. Развивая терминологию «using-implementing» [3], соответствующие уровни можно обозначить как «formal using - формальное использование» (уровень 1), «using&implementing - использование и знание реализации» (уровень 2), «using&implementing&analyzing - использование, реализация и анализ» (уровень 3).

На уровне бакалавриата реально говорить только о возможности достижения первых двух уровней, причем не всеми студентами. Например, J. Adams [3] приводит данные о том, что обязательный курс 2 семестра «Introduction To Data Structures», в котором в дополнение к «стандартным» темам рассматриваются программирование потоков, простейшие параллельные алгоритмы и их реализация с помощью OpenMP C++, изучало 40 студентов направления Computer Science. А итоговый элективный курс «High Performance Computing» в 7 семестре слушало 8 студентов. В условиях жестких отечественных учебных планов об аналогичной тенденции свидетельствуют оценки студентов.

Поэтому для таких направлений, как «Математика и компьютерные науки», «Прикладная информатика», «Информационные системы и технологии» и т.д. представляется оптимальным чтение в 7 семестре интегрированного курса, включающего обзор основных сведений об архитектуре многопроцессорных систем, моделях и технологиях параллельного программирования. Опыт чтения такого курса позволяет констатировать, что он (точнее, освоение его содержания) является своеобразной лакмусовой бумажкой, показывающей не только то, на каком уровне изучались и как были усвоены студентами базовые понятия архитектуры компьютеров, операционных систем, сетевых технологий, семантики языков программирования,

алгоритмов компьютерной обработки данных. Не менее важно следующее: изучение основ НРС не является для студентов жизненно важной задачей, поэтому именно здесь очень четко проявляется то, кто из них хочет и способен учиться новому.

Для компактного интегрированного курса очень важно особое информационное обеспечение. Изучение опыта ведущих университетов России, в течение продолжительных сроков реализующих подготовку в области НРС (МГУ, НГУ, ННГУ, СПбГУ, ЮУрГУ и др.), показало, что, как правило, студентам предоставляются краткие учебные пособия и/или презентации лекций по основным технологиям параллельного программирования (MPI, OpenMP), а также задания для самостоятельной работы, практически однотипные для большинства вузов и учебных курсов (независимо от их продолжительности и аудитории). От обучающихся, как правило, требуется протестировать уже готовую программу (решение системы линейных алгебраических уравнений, численное интегрирование и т.п.) и провести вычислительный эксперимент для анализа ускорения и эффективности в нескольких режимах на обычном 2-х или 4-ядерном процессоре (в зависимости от возможностей доступного компьютера), а также на учебной или профессиональной многопроцессорной вычислительной системе.

Возможно, для крупных федеральных и национальных исследовательских университетов, тесно интегрированных с научно-исследовательскими институтами, такой самостоятельной работы вполне достаточно для того, чтобы студенты закрепили некие общие представления о предметной области и основных подходах параллельного программирования. При этом основное освоение технологий происходит уже в процессе активной исследовательской деятельности, индивидуальной работы со студентами в рамках лабораторий и научно-образовательных центров. Но поскольку суперкомпьютерные технологии получают все большее распространение, встает вопрос о том, как реализовывать соответствующие образовательные программы и отдельные курсы для студентов различных направлений подготовки в «обычных» вузах, используя широкие возможности современного информационного пространства.

Мы исходили из того, что для успешного освоения любых новых технологий студентам нужно четко видеть и уровень мировых достижений в соответствующей области, и международные стандарты требований к подготовке специалистов. Именно это и послужило основным принципом отбора информационных ресурсов для интегрированного курса. Наряду с авторскими материалами (краткими конспектами, презентациями лекций и заданиями для самостоятельной работы, разработанными преподавателями университета) студентам предлагались ссылки на презентации и видеозаписи лекций Интернет-университета суперкомпьютерных технологий ([www.hpcu.ru](http://www.hpcu.ru)), лекции и тесты Интернет-университета информационных технологий ([www.intuit.ru](http://www.intuit.ru)), лабораторные работы Сибирского суперкомпьютерного центра для студентов НГУ, методические материалы и лабораторные работы для студентов МГУ, ННГУ, СПбГУ, а также ряд других открытых материалов российских вузов.

Особое место среди предлагаемых студентам ресурсов занимают аутентичные технические руководства и описания стандартов технологий НРС от фирм-вендоров и консорциумов (Cisco Systems, Intel, Microsoft, OpenMP Architecture Review Board и др.). Наконец, очень важен разбор на практических занятиях заданий, предлагаемых студентам ведущих зарубежных университетов именно в текущем семестре. Например, в осеннем семестре 2011-2012 учебного года студенты 4-ого курса направления «Математика и компьютерные науки» на практических занятиях по интегрированному курсу «Параллельное программирование» под руководством преподавателя разбирали задания для лабораторных работ по теме «Shared Memory Programming With OpenMP» в рамках курса «Advanced Computational Science II», ведущимся в это же время департаментом «Scientific Computing» Florida State University ([people.sc.fsu.edu/~jburkardt/](http://people.sc.fsu.edu/~jburkardt/)).

Иной характер носят содержание и формы подготовки в области НРС для таких специальностей как «Компьютерная безопасность» и «Информационная безопасность автоматизированных систем», реализуемых в ТюмГУ, поскольку здесь в условиях только включенных разделов необходимо добиться уровня 3 («using&implementing&analyzing - использование, реализация и анализ»). Для иллюстрации остановимся на кратком обзоре содержания включенного раздела «Технологии параллельного программирования» дисциплины «Языки программирования», которая читается во 2-4 семестрах. В качестве основы используются программирование на C/C++ и ранее рассмотренные модели реализации подпрограмм, а также управления памятью. Именно для их иллюстрации в данном разделе (4 семестр) рассматриваются технологии OpenMP и MPI с упором на особенности реализации соответствующих управляющих конструкций и функций. По нашему мнению заслуживает внимания тот факт, что студенты в итоге самостоятельно приходят к мысли о возможности гибридного параллельного программирования MPI/OpenMP, после чего им предлагаются для изучения обзорные электронные ресурсы (например, [4]). Также в этом разделе в развитие представлений об использовании базовых классов .NET и делегатов (практическое программирование на C++ и C#) изучаются классы System.Threading.Thread, System.Threading.Tasks с особым вниманием к особенностям реализации методов Parallel.Invoke, Parallel.For, Parallel.ForEach. Включение такого раздела не только не разрушает общей концепции курса – изучение семантики языков программирования, но и позволяет органично рассмотреть и сопоставить сущность нескольких моделей и технологий параллельного программирования.

Естественно, что успешное изучение включенных разделов требует значительного объема индивидуальной работы со студентами. Она проводится, в основном, благодаря энтузиазму преподавателей, который, в свою очередь, поддерживается заинтересованностью наиболее сильных студентов. Поэтому по

умолчанию приходится считать эти темы факультативными для большинства. Вот возможный ответ на один поставленных вопросов (обязательная или факультативная дисциплина): составной факультатив, интегрированный в обязательный курс.

Таким образом, нет никаких помех для включения в содержание образования соответствующей предметной области. Поэтому, как правило, то, в какой форме и в каком объеме вопросы НРС затрагиваются в учебном плане, целиком зависит от его разработчиков, а содержание соответствующих курсов определяется интересами и профессиональным уровнем преподавателей.

#### ЛИТЕРАТУРА:

1. Проект реализации объединенной «Национальной суперкомпьютерной технологической платформы» [электронный ресурс] / Режим доступа: <http://www.hpc-platform.ru/tiki-index.php?page=basic-docs>, свободный.
2. Adams J., Nevison C., Schaller N.C. Parallel Computing to Start the Millennium// ACM SIGCSE Bulletin, Volume 32 Issue 1, Mar. 2000. P. 65-69.
3. Adams J. Integration of Parallel Topics in the Undergraduate Curriculum [электронный ресурс] / Режим доступа: <http://techtalks.tv/talks/integration-of-parallel-topics-in-the-undergraduate-curriculum-2011/5306/>, свободный.
4. Hanke M. Combining OpenMP and Message Passing: Hybrid Programming [электронный ресурс] / Режим доступа: <http://www.pdc.kth.se/education/historical/previous-years-summer-schools/2009/handouts/lect6.pdf>, свободный.