

ИСПОЛЬЗОВАНИЕ ИНТЕРНЕТА ДЛЯ ОБУЧЕНИЯ ПАРАЛЛЕЛЬНОМУ ПРОГРАММИРОВАНИЮ

**В.А. Бахтин, О.Ф. Жукова, Н.А. Катаев, А.С. Колганов, В.А. Крюков,
Н.В. Поддерюгина, М.Н. Притула, О.А. Савицкая, А.А. Смирнов**

Институт прикладной математики им. М.В. Келдыша РАН

Количество пользователей, ежедневно входящих в сеть Интернет, активно растет. По данным Всероссийского центра изучения общественного мнения (ВЦИОМ) доля интернет-пользователей в 2015 году составляет 69%, а доля тех, кто выходит в Сеть ежедневно, составляет половину россиян (52%). Доля интернет-пользователей среди молодежи (18-24 летних) составляет 96%. Самыми популярными поводами обращения к Сети являются медиа-ресурсы (56%), поиск новостей и информации (55%), электронная почта (53%), работа и учеба (50%). Опрос проведен 21-22 марта 2015 года, опрошено 1600 человек в 130 населенных пунктах в 46 областях, краях и республиках России [1]. Результаты опроса показывают, что, поскольку все большее количество молодых людей пользуются Интернетом, Всемирная сеть становится важным ресурсом для использования в сфере образования. Это ставит перед преподавателями ряд вопросов. Нужно, чтобы информационная среда была не просто доступна студентам, но и удовлетворяла следующим требованиям. Прежде всего, она должна содержать достоверную информацию, включать общепринятую терминологию, подачу информации следует осуществлять от простого к сложному, облегчая тем самым усвоение ее студентами. Это должно сопровождаться укреплением связи студент-преподаватель, в том числе и для усиления контроля преподавателя за образовательным процессом.

Активное использование подхода Веб 2.0 к организации Интернет-ресурсов ведет к тому, что бесконтрольный поиск информации приводит студентов на ресурсы, достоверность которых может вызывать сомнения. Следует обратить внимание, что в интернет-энциклопедии Википедия на странице, озаглавленной «Отказ от ответственности» [2], крупными буквами указано «Википедия не гарантирует истинности». Это связано с тем, что приведенные в ней данные «могли быть только что изменены, испорчены, либо написаны тем, чье мнение отличается от общепринятого в области знаний, интересующей вас».

Кроме того, поиск нужной информации может сам по себе вызывать значительные трудности, связанные с колоссальным объемом данных, составляющих интернет-пространство [3]. Огромную роль играет формулировка запроса, правильно указанные ключевые слова. Если поиск проводится в рамках специализированной области, такой как параллельное программирование, важную роль играет знание общепринятой терминологии, в

том числе на иностранных языках. В качестве примера рассмотрим исследования, проводимые в области статического анализа программ. Результаты анализа применяются в автоматически распараллеливающих компиляторах, в системах автоматизированного распараллеливания, таких как САПФОР [4]. Для эффективного распараллеливания циклов последовательной программы необходимо указать, какие переменные могут быть объявлены приватными. В терминологии OpenMP [5] такие переменные должны быть перечислены в директивах `private`. Поиск по запросу `private variables` в поисковой системе Google не дает нужных результатов, так как аналогичная терминология используется при ограничении области видимости во многих языках программирования, не связанных с параллелизмом. Указав вместо `private` – `privatizable`, что точнее соответствует цели поиска (переменные, которые могут быть использованы как приватные) и, добавив ключевое слово `analysis`, можно найти статьи, посвященные соответствующим алгоритмам, применяемым при статическом анализе программ.

Справиться с перечисленными трудностями можно, подготовив ссылки на электронные ресурсы с проверенной информацией по изучаемой проблеме, подготовив ресурсы, описывающие задания и контролирующие их выполнение, наладив электронную переписку, в том числе организовав списки рассылки. Возможно привлечение различных средств связи через Интернет в режиме реального времени (например, Skype).

При обучении параллельному программированию особое внимание следует уделить доступу к вычислительным ресурсам. Суперкомпьютерные системы доступны только удаленно и, как правило, сильно загружены. Это не позволяет оперативно выполнять большое количество практических заданий, необходимых в процессе обучения. Наряду с доступом к мощным вычислительным комплексам, доступ к которым необходим для полного понимания изучаемой области, необходимо организовывать доступ к небольшим рабочим станциям для практического совершенствования навыков параллельного программирования.

Спецсеминар «Операционные системы и языки программирования распределенных вычислительных систем» кафедры Системного программирования (СП) факультета Вычислительной математики и кибернетики (ВМиК) Московского государственного университета им М.В. Ломоносова (МГУ) [6] проводит подготовку в области суперкомпьютерных технологий. Работа спецсеминара связана с исследованиями, проводимыми в направлениях разработки высокоуровневых языков параллельного программирования и создания систем автоматизированного распараллеливания. Студенты получают теоретическую подготовку и приобретают опыт разработки параллельных программ для современных вычислительных кластеров. Исследовательская деятельность студентов и аспирантов направлена на разработку инструментов параллельного программирования. При их активном участии развиваются системы DVM [7, 8]

и САПФОР [4], разработанные в Институте прикладной математики им. М.В. Келдыша РАН.

Обучение программированию невозможно без выполнения практических заданий, и параллельное программирование – не исключение. Система контроля выполнения практических заданий организована на базе веб-приложения для управления проектами Redmine [9]. Прямое назначение данного инструмента – управление проектами, задачами, отслеживание ошибок. Инструмент используется при разработке систем DVM и САПФОР для распределения задач между разработчиками, для отслеживания статуса задач и ошибок, обнаруживаемых в системе, для расставления приоритетов задач. Каждой из разрабатываемых систем в Redmine соответствует отдельный проект.

При обучении параллельному программированию создается учебный проект, в котором для студентов распределяются задания. Таким образом, студенты одновременно с изучением параллельного программирования приобретают навыки по управлению проектами, и для привлечения их к участию в разработке систем DVM или САПФОР не требуется дополнительного изучения данного инструмента.

Каждый студент, участвующий в спецсеминаре, регистрируется в системе Redmine, при этом создается аккаунт, связанный с почтовым адресом студента. В процессе работы над проектом на указанный адрес приходит информация о статусе выполняемых студентом задач, отправляются оповещения о появлении новых задач, о том, считается ли задача сданной. Система Redmine позволяет отвести каждому участнику проекта определенную роль, необходимым образом ограничив доступ к функциям системы. Каждый студент получает роль Student. Затем ему будет поставлена отдельная задача, которую он должен выполнить. В каждом проекте создается страница Wiki с описанием общей части поставленных задач (см. Рис. 1).

Преподаватели спецсеминара также зарегистрированы в системе и участвуют в учебном проекте в роли Manager. С каждой задачей связан аккаунт одного или нескольких преподавателей, отслеживающих ее выполнение. При изменении состояния задачи на электронные адреса преподавателей приходят соответствующие уведомления.

Выполнение задачи происходит в несколько этапов, сложность каждого этапа постепенно возрастает. Каждый этап оформляется в системе как подзадача. На каждом этапе студент последовательно знакомится с различными технологиями параллельного программирования (OpenMP, OpenACC [10], DVMH), а также изучает средства профилирования программ, такие как Intel VTune Amplifier [11].

Home My page Projects Administration Help Logged in as N_Kataev My account Sign out

Practical training 2014 Search: Practical training 2014 ▼

Overview Activity Issues New issue Gantt Calendar **Wiki** Files Settings

Задание 2014 г.

[Edit](#) [Watch](#) [Lock](#) [Rename](#) [Delete](#) [History](#)

В 2014 году каждому студенту выдаются в качестве исходных две Фортран-программы – последовательная версия одного из тестов NAS NPB 3.3.1 и ее параллельная версия в модели OpenMP. Требуется распараллелить последовательную программу в моделях OpenACC и DVMN. Это задание разбивается на следующие этапы, каждому из которых в итоговом отчете соответствует раздел.

- 1. Получение на целевой ЭВМ времен выполнения**

Получение на целевой ЭВМ (например, K-100, Ломоносов или др.) для нескольких классов начальных данных (классы A, C) следующих времен выполнения:

 - время выполнения последовательной версии;
 - время последовательного выполнения OpenMP-версии;
 - время выполнения OpenMP-версии на 1, 2, 4, 8, 12 нитях одного узла.
- 2. Профилерование выполнения последовательной программы**

Профилерование последовательной и OpenMP-программы может осуществляться при помощи:

 - Intel VTune Amplifier XE 2013
 - Intel Advisor XE

Данные инструменты входят в состав Intel Cluster Studio XE 2013, Intel Parallel Studio XE или Intel Fortran Studio XE. Пробные версии для Windows (на 30 дней) или версию для некоммерческого использования для Linux можно скачать с сайта <http://software.intel.com/ru-ru/>.

Для профилерования OpenMP-программы на K-100 можно использовать опцию `-openmp-profile`. В результате запуска программы, скомпилированной с опцией `-openmp-profile`, на выполнение создается текстовый файл `guide.gvs` - программа разбивается на последовательные и параллельные участки, и для каждого такого участка собираются характеристики эффективности.

Также для профилерования может быть полезен AMD CodeXL: <http://developer.amd.com/tools-and-sdks/heterogeneous-computing/codexl/>
- 3. Профилерование выполнения OpenMP-версии на 1, 2, 4, 8, 12 нитях**
- 4. Распараллеливание программы на OpenACC и получение на целевой ЭВМ времен выполнения на одном GPU**
- 5. Распараллеливание программы на DVM и получение на целевой ЭВМ времен выполнения**

Распараллеливание программы на DVM и получение на целевой ЭВМ времен выполнения на 1, 2, 4, 8, 12 процессах одного узла, а также времени выполнения этой DVM-программы в последовательном режиме.
- 6. Распараллеливание программы на DVMN и получение на целевой ЭВМ времен выполнения**

Распараллеливание программы на DVMN и получение на целевой ЭВМ времен выполнения на 1,2,3 GPU.

Wiki
[Start page](#)
[Index by title](#)
[Index by date](#)

Рис. 1. Пример задания, выполняемого студентами в 2014 году

На Рис. 2 приведен пример отображения состояния задачи по распараллеливанию теста из пакета NAS NPB [12]. Основная задача состоит из шести подзадач, соответствующих определенным этапам обучения. Для студенческих задач допускается четыре стадии выполнения. *New* означает, что задача только создана и студент еще не приступил к ее выполнению. *In Progress* устанавливается студентом для решаемых в данный момент задач. *Resolved* устанавливается студентом, когда он считает, что задача выполнена. Статус *Closed* может быть установлен только преподавателем после проверки корректности выполнения задачи. В процессе выполнения студент указывает время, которое он затратил на каждый этап задания и, если задача еще не завершена, процент от ее полной готовности. Когда задача считается законченной, к ее описанию добавляется отчет о выполнении (см. Рис. 3).

#	Tracker	Status	Priority	Subject	Updated	% Done
232	Feature	In Progress	Normal	Распараллеливание теста NAS MG на OpenACC и DVMH	22.12.2014 22:10	<div style="width: 100%;"></div>
233	Support	Closed	Normal	MG Получение на целевой ЭВМ времен выполнения	25.05.2015 23:22	<div style="width: 100%;"></div>
234	Support	Closed	Normal	MG Профилирование выполнения последовательной программы	25.05.2015 23:23	<div style="width: 100%;"></div>
235	Support	Resolved	Normal	MG Профилирование выполнения OpenMP-версии на 1, 2, 4, 8, 12 нитях	28.10.2014 09:07	<div style="width: 100%;"></div>
236	Feature	Resolved	Normal	MG Распараллеливание программы на OpenACC и получение на целевой ЭВМ времен выполнения на одном ГПУ	02.12.2014 16:52	<div style="width: 100%;"></div>
237	Feature	In Progress	Normal	MG Распараллеливание программы на DVM и получение на целевой ЭВМ времен выполнения	22.12.2014 22:10	<div style="width: 100%;"></div>
238	Feature	New	Normal	MG Распараллеливание программы на DVMH и получение на целевой ЭВМ времен выполнения	10.09.2014 12:11	<div style="width: 100%;"></div>

Рис. 2. Пример отображения состояния выполняемой задачи

Support #233

Feature #232: Распараллеливание теста NAS MG на OpenACC и DVMH
MG Получение на целевой ЭВМ времен выполнения

Added by Nikita Kataev 9 months ago. Updated 14 minutes ago.

Status: Closed
Priority: Normal
Assignee: [Progress Bar]
Category: -
Target version: -

Start date: 01.10.2014
Due date: 07.10.2014
% Done: 100%
Estimated time: 16.00 hours

Description

Получение на целевой ЭВМ (например, K-100, Ломоносов или др.) для нескольких классов начальных данных (классы A, C) следующих времен выполнения:

- время выполнения последовательной версии;
- время последовательного выполнения OpenMP-версии;
- время выполнения OpenMP-версии на 1, 2, 4, 8, 12 нитях одного узла.

Subtasks

Related issues

History

Updated by Nikita Kataev 9 months ago #1

- copied from Support #219: VT Получение на целевой ЭВМ времен выполнения added

Updated by Nikita Kataev 9 months ago #2

- copied from deleted (Support #219: VT Получение на целевой ЭВМ времен выполнения)

Updated by Nikita Kataev 9 months ago #3

- Subject changed from VT Получения на целевой ЭВМ времен выполнения to MG Получение на целевой ЭВМ времен выполнения
- Assignee changed from [User] to [User]

Updated by [User] 8 months ago #4

- Status changed from New to Resolved
- % Done changed from 0 to 100

Получены времена выполнения теста MG на K-100:

CLASS	Последовательный вариант	Последовательный OpenMP вариант	OpenMP 1 процесс	OpenMP 2 процесса	OpenMP 4 процесса	OpenMP 8 процессов	OpenMP 12 процессов
A	1.41s	1.41s	1.56s	0.83s	0.42s	0.31s	0.33s
C	55.71s	55.38s	60.48s	31.01s	18.42s	13.87s	13.38s

Рис. 3. Подробное описание задачи

На определенном этапе обучения студенты приступают к изучению DVM-системы для разработки параллельной версии программы для кластера с графическими ускорителями. Распараллеливание выполняется с помощью DVMH-расширения стандартных языков параллельного программирования Fortran или C. Знакомство с DVM-системой начинается с обращения к сайту данного проекта [13], созданного авторами статьи.

Цель создания сайта DVM-системы – предоставить пользователям разного уровня подготовки, в том числе студентам, изучающим параллельное программирование, актуальную информацию о возможностях системы, заинтересовать и привлечь новых пользователей, обеспечить удобный доступ к системе и оперативную обратную связь с пользователями. Доступны русскоязычная и англоязычная версии сайта. Выделяется пять основных разделов: главная страница, документация, получение доступа к системе, публикации и доклады, презентующие различные особенности системы, контактная информация.

Попадая на сайт, пользователь получает представление о целях создания DVM-системы, о задачах, на решение которых она ориентирована, о том, что такое DVMH-модель параллельного программирования, о существующих средствах поддержки процесса разработки параллельных программ с использованием DVMH-модели. Для начального ознакомления с ключевыми возможностями языков Fortran-DVMH и C-DVMH доступны подробно разобранные примеры программ. Примеры могут быть полезны как студентам, только начинающим изучать возможности DVM-системы так и пользователям, обладающим базовым уровнем знаний в области параллельного программирования. Для углубленного изучения DVMH-модели подготовлено отдельное руководство, ознакомившись с которым можно выполнить распараллеливание программ разного уровня сложности. Указанное руководство используется студентами при выполнении практических заданий на этапе разработки DVMH-версий последовательных программ. Проследить историю развития DVM-системы и изучить отдельные аспекты DVMH-модели можно в разделе публикаций и докладов, в котором наряду с текстами статей, опубликованными в различных журналах и материалах конференций, доступны презентации и видео с выступлений разработчиков DVM-системы на конференциях, посвященных параллельному программированию.

Получить доступ к DVM-системе можно тремя способами: установить систему на своем компьютере, получить удаленный доступ к рабочей станции или воспользоваться суперкомпьютерами, на которых система предустановлена. Персональные компьютеры и рабочие станции оптимальны для начального обучения технологиям параллельного программирования. На них нет очередей задач и ограничений по запуску программ. При этом на начальном этапе не требуется большого количества вычислительных ресурсов. Чтобы получить доступ к рабочей станции с установленной DVM-системой, необходимо заполнить форму доступа на сайте. Также на сайте доступны инструкции по использованию системы на суперкомпьютерах «Ломоносов» (НИВЦ МГУ), K100 (ИПМ им. М.В. Келдыша), МВС-10П (МСЦ РАН), МВС-100k (МСЦ РАН).

На основе вышесказанного можно сделать следующий вывод. Интернет-технологии позволяют обеспечить оперативную связь студентов с преподавателем, постоянный контроль за ходом образовательного процесса,

доступ к Интернет-ресурсам, содержащим актуальную и оперативно обновляющуюся информацию по соответствующей тематике. При обучении суперкомпьютерным технологиям для практической работы необходимо иметь доступ к вычислительным ресурсам, использование которых возможно только удаленно. Необходимо учитывать, что Интернет-ресурсы предоставляют доступ к обширному набору неструктурированной информации, достоверность источников, которой может вызывать сомнения.

В статье было рассмотрено применение сетевых технологий в процессе обучения параллельному программированию на факультете ВМиК МГУ им. М.В. Ломоносова. Студенты получают навыки удаленной работы с вычислительными ресурсами, знакомятся с правилами работы на суперкомпьютерных системах, обучаются работе с системами очередей для запуска своих программ. Обучение включает в себя знакомство с технологиями параллельного программирования MPI, OpenMP, OpenACC и DVMH. Авторами статьи был разработан сайт DVM-системы, содержащий подробное руководство по использованию системы, примеры программ, публикации и доклады. Была организована система контроля процесса обучения на базе веб-приложения для управления проектами.

ЛИТЕРАТУРА

1. Пресс-выпуск №2836 [Электронный ресурс].
URL: <http://wciom.ru/index.php?id=236&uid=115255> (дата обращения 14.10.2015).
2. Википедия: Отказ от ответственности [Электронный ресурс].
URL: http://ru.wikipedia.org/wiki/Википедия:Отказ_от_ответственности (дата обращения 14.10.2015).
3. Цымблер М.Л. Big Data: несколько простых вопросов о сложном явлении // Суперкомпьютеры 1(17) весна 2014 — ООО «Издательство СКР Медиа», С. 8-11.
4. Бахтин В.А., Бородич И.Г., Катаев Н.А., Клинов М.С., Ковалева Н.В., Крюков В.А., Поддерюгина Н.В. Диалог с программистом в системе автоматизации распараллеливания САПФОР // Вестник Нижегородского университета им. Н.И. Лобачевского. — 2012 — № 5 (2). — С. 242–245.
5. OpenMP Application Program Interface Version 4.0 - July 2013 [Электронный ресурс].
URL: <http://www.openmp.org/mp-documents/OpenMP4.0.0.pdf> (дата обращения 14.10.2015).
6. Спецсеминар «Операционные системы и языки программирования распределенных вычислительных систем» [Электронный ресурс].
URL: <http://sp.cs.msu.ru/specsem/ospl/> (дата обращения 14.10.2015).
7. Коновалов Н.А., Крюков В.А., Михайлов С.Н., Погребцов А.А. Fortran DVM - язык разработки мобильных параллельных программ // Ж. "Программирование", № 1, 1995, С. 49-54.

8. Бахтин В.А., Клинов М.С., Крюков В.А., Поддерюгина Н.В., Притула М.Н., Сазанов Ю.Л. Расширение DVM-модели параллельного программирования для кластеров с гетерогенными узлами. – Вестник Южно-Уральского государственного университета, серия "Математическое моделирование и программирование", №18 (277), выпуск 12 – Челябинск: Издательский центр ЮУрГУ, 2012, с. 82-92.
9. Overview – Redmine [Электронный ресурс].
URL: <http://www.redmine.org/> (дата обращения 14.10.2015)
10. OpenACC Home | www.openacc.org [Электронный ресурс].
URL: <http://www.openacc.org/> (дата обращения 14.10.2015).
11. Intel VTune Amplifier | Intel Developer Zone [Электронный ресурс]. URL: <https://software.intel.com/en-us/intel-vtune-amplifier-xe> (дата обращения 14.10.2015).
12. NAS Parallel Benchmarks [Электронный ресурс].
URL: <http://www.nas.nasa.gov/publications/npb.html> (дата обращения 14.10.2015).
13. DVM-система | Система разработки параллельных программ [Электронный ресурс].
URL: <http://dvm-system.org/> (дата обращения 14.10.2015).