

Министерство образования и науки Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра математического обеспечения вычислительных комплексов и
информационных систем

**СРАВНЕНИЕ ЭФФЕКТИВНОСТИ РАСПАРАЛЛЕЛИВАНИЯ
ВЫЧИСЛЕНИЙ НА АРХИТЕКТУРАХ x86_64 и INTEL MIC
МОДЕЛИРОВАНИИ НЕЛИНЕЙНЫХ КОМБИНИРОВАННЫХ
ДИНАМИЧЕСКИХ СИСТЕМ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы

направления 02.03.03 Математическое обеспечение и администрирование
информационных систем

факультета компьютерных наук и информационных технологий

Хижнякова Арсения Андреевича

Научный руководитель:

Доцент, д.ф.-м. н. _____

Д. К. Андрейченко

подпись, дата

Зав. кафедрой:

Доцент, д. ф.-м. н _____

Д. К. Андрейченко

подпись, дата

Саратов 2016

ВВЕДЕНИЕ

Актуальность темы. В начале эры развития РС (Персонального компьютера) возможности, которые предоставлялись пользователю, были крайне малы – это была простая однопользовательская система, которая могла выполнять только одну программу в каждый момент времени. И программы эти следовали прямолинейному выполнению команд.

С течением времени экспоненциальный рост вычислительной мощности привёл к созданию более мощных систем, а соответственно и созданию более совершенной пользовательской среды. Такой рост повлёк за собой завышенные ожидания пользователей – они хотели бы иметь возможность посылать электронную почту, и одновременно с этим слушать потоковое аудио из интернет – радиостанции, более быстрый отклик и работу программ. Это и есть основные проблемы, стоящие сегодня перед разработчиками программного обеспечения.

Параллельное программирование – это область, которая связана с организацией расчётов на системах, состоящих из нескольких процессорных устройств, таких как многоядерные процессоры.

На современном этапе развития параллельные информационные технологии являются необходимой составляющей комплекса знаний разработчика программного обеспечения.

Цель бакалаврской работы: сравнение эффективности моделирования нелинейных комбинированных динамических систем на архитектурах x86_64 и Intel MIC, применительно к задаче моделирования цилиндрического гидродинамического подвеса с учётом квадратичной нелинейности в уравнениях Навье–Стокса поддерживающего слоя жидкости.

Задачи бакалаврской работы:

1. Изучить особенности работы с архитектурами x86_64 и Intel MIC.
2. Пересобрать и адаптировать проект, написанный под ОС Windows, для архитектур x86_64 и Intel MIC.

3. Сравнить эффективность распараллеливания типовых вычислений на основе технологии OpenMP и с использованием кластера с сопроцессором ускорителем IntelXeonPhi. В качестве основного примера берётся реакция на скачок Хевисайда.

Методологические основы адаптации проекта под предложенные архитектуры приводятся в источниках [14,15,16,17]. Методологические основы самой модели приводятся в источниках [12,13].

Практическая значимость бакалаврской работы. В ходе выполнения практической части бакалаврской работы был изучен способ пересборки и адаптации проекта, написанного на ОС Windows, для архитектур x86_64 и IntelMIC, а так же способы работы с кластером и вычисления на нём задач. Материал может быть использован как пример для адаптации проекта под другую архитектуру.

Структура и объем бакалаврской работы. Работа состоит из введения, 5 разделов, заключения, списка использованных источников информации и 6 приложений. Общий объем работы – 52 страницы,

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Симметричные мультипроцессорные системы с общей памятью и технология OpenMP» посвящен рассмотрению технологии параллельного программирования OpenMP. Описаны её директивы и функции, а также рассмотрены примеры работы с ней.

Второй раздел «Архитектура IntelMIC» описывает особенности и поколения данной архитектуры, а также в разделе рассмотрен сопроцессор ускоритель IntelXeonPhi, на котором так же производились вычисления.

Третий раздел «Комплексные динамические системы» посвящен краткому описанию математической модели.

Четвёртый раздел «Сравнение эффективности технологий на примере КДС» описывает саму математическую модель, с применением формул и пояснений.

Пятый раздел «Результаты моделирования» содержит описание пересборки и адаптации проекта, с приведением скриншотов, а также результаты моделирования тестовой математической модели на данных архитектурах.

Технология OpenMP является наиболее популярным средством программирования для симметричных мультимикропроцессорных систем с общей памятью, к которым и относится подавляющее число серийным образом выпускаемых в настоящее время ноутбуков, рабочих станций и серверов.

Программа представляет собой чередующуюся последовательность последовательных и параллельных областей. При входе в параллельную область из последовательной, начальный поток становится основным, а запускаемые исполнительной системой OpenMP дополнительные потоки наряду с основным распределяют между собой вычислительную работу. При выходе из параллельной области в последовательную, основной поток снова приобретает статус начального, а дополнительные потоки завершаются (фактически – приостанавливают свою работу). При повторном входе в параллельную область снова запускаются (фактически – возобновляют свою работу) дополнительные потоки. И так далее. Теоретически, если вычислительная работа равномерно распределяется между потоками, а число потоков в параллельной области совпадает с числом имеющихся вычислителей (процессоров, ядер и т.д.), то время выполнения параллельной области сокращается пропорционально числу вычислителей. Однако, поскольку при реализации тех или иных алгоритмов распараллелить удастся не все операции (т.е. наряду с параллельными областями будут существовать и последовательные), общее ускорение вычислительного процесса может оказаться несколько меньше теоретического [3].

Важным критерием в создании параллельного кода является загрузка, синхронизация и балансировка потоков в OpenMP. При неудачном решении данных проблем все усилия по созданию параллельной версии программы

могут быть напрасны. Параллельная версия программы может оказаться медленнее последовательной версии.

Состояние гонки – ошибка проектирования или реализации многозадачной системы, при которой работа системы зависит от того, в каком порядке выполняются части кода. Эта разновидность ошибки является наиболее распространенной при параллельном программировании и весьма коварна. Воспроизведение и локализация этой ошибки часто бывает затруднена в силу непостоянства своего проявления.

Технология нацелена на то, чтобы пользователь имел один вариант программы для последовательного, а также для параллельного выполнения. За основу в OpenMP берётся программа, для создания параллельной версии которой пользователю предоставляется набор директив, функций и переменных. Однако, результат выполнения команды в параллельном режиме будет отличаться от результата, выполненного в последовательном.

OpenMP поддерживает работу с языками C/C++/Фортран. Первая спецификация для языка Фортран появилась в октябре 1997 года, для C/C++ в октябре 1998 года.

Реализация параллельных вычислений в OpenMP достигается с помощью многопоточности, в которой главный поток, называемый “master” создаёт набор подчиненных, называемых “slave”.

Важным достоинством технологии OpenMP является возможность реализации так называемого инкрементального программирования, когда программист постепенно находит участки в программе, содержащие ресурс параллелизма, с помощью предоставляемых механизмов делает их параллельными, а затем переходит к анализу следующих участков. Таким образом, в программе не распараллеленная часть постепенно становится всё меньше. Такой подход значительно облегчает процесс адаптации последовательных программ к параллельным компьютерам, а также отладку и оптимизацию.

Компиляция программы: для того, чтобы использовать механизм

OpenMP необходимо скомпилировать программу компилятором, который поддерживает OpenMP, с указанием соответствующего ключа (в VisualC++ /**openmp**). После этого компилятор интерпретирует директивы OpenMP и создаёт параллельный код.

IntelMIC – разработанная компанией Intel архитектура многоядерной процессорной системы. Прототип процессоров архитектуры MIC был выпущен в 2010 году. В 2012 году компания Intel объявила о ребрендинге процессоров под названием XeonPhi.

Продукты на базе архитектуры IntelMIC предназначены для областей и систем, которые интенсивно используют параллельную обработку данных. Высокая степень параллелизма в данной архитектуре достигается за счет использования процессорных ядер меньшего размера, которые в свою очередь потребляют меньше энергии. Как результат высокая производительность в системах с интенсивной параллельной обработкой данных [8].

IntelXeonPhi – сопроцессор, имеющий формат дополнительного адаптера и работающий совместно с процессорами IntelXeon, обеспечивая резкий прирост производительности при высокопараллельной обработке кода. Прирост составляет до 1.2 терафлопс (триллион операций с плавающей запятой в секунду) двойной точности на сопроцессор.

Каждый сопроцессор имеет больше ядер, больше вычислительных потоков и больше исполнительных блоков векторных команд, чем процессор IntelXeon. Более низкая скорость вычислений в каждом ядре компенсируется благодаря высокому уровню параллелизма, что обеспечивает высокую совокупную производительность при выполнении высокопараллельных вычислений.

Для того, чтобы полностью раскрыть возможности сопроцессоров IntelXeonPhi, приложение должно масштабироваться до уровня более 100 программных потоков, либо активно использовать векторные вычисления. Области применения высокопараллельной обработки: анимация, финансы,

медицина, производство. Из дополнительных преимуществ IntelXeonPhi выделяют единую модель программирования для всего кода – IntelXeonPhi предоставляет разработчику аппаратный модуль, который оптимизирован для высокого уровня параллелизма и который не требует изменения архитектуры приложения или переписывания кода. Процессоры IntelXeon и сопроцессоры IntelXeonPhi поддерживают единый исходный код, следовательно, разработчикам необходимо провести лишь однократную оптимизацию приложений для параллельных вычислений, чтобы добиться максимальной производительности как процессора, так и сопроцессора. Благодаря тому, что сопроцессор разработан с нуля для высокопроизводительных вычислений, в отличие от графических процессоров сопроцессор может иметь собственную ОС, полностью поддерживать IP– адресацию и стандарт MPI [9,10,11].

PuTTY— свободно распространяемый клиент для различных протоколов удалённого доступа. Позволяет подключиться и управлять удалённым узлом. В PuTTY реализована только клиентская сторона соединения — сторона отображения, в то время как сама работа выполняется на другой стороне. Исходный код PuTTY полностью разработан на C. PuTTY не зависит от DLL, других приложений, пакетов обновлений ОС. Пакет состоит только из исполняемых файлов, которые могут быть установлены в любом месте. Программа является свободно распространяемым приложением с открытым исходным кодом и выпускается под OpenSource лицензией MIT.

Для организации доступа к файловой системе сервера(кластера) используется программа WinSCP — свободный графический клиент протоколов SFTP и SCP. Обеспечивает защищённое копирование файлов между компьютером и серверами, поддерживающими эти протоколы. Для подключения к соответствующему серверу при первоначальном запуске программы задаётся адрес сервера, а также логин и парольная фраза, созданные при работе с PuTTY [15].

Аккаунту соответствует папка на узле кластера, в которую можно загружать необходимые исполняющие файлы, а затем с помощью PuTTY запускать их на исполнение на сервере.

ЗАКЛЮЧЕНИЕ

В ходе выполнения бакалаврской работы были выполнены все поставленные задачи, а именно изучены особенности работы с архитектурами x86_64 и IntelMIC, пересобран и адаптирован проект в виде математической модели, и произведено сравнение эффективности распараллеливания вычислений на данных архитектурах.

Для работы с сервером были выбраны программы WinSCP и PuTTY, с которыми легко и удобно работать, благодаря интуитивно понятному интерфейсу и наличию множества справок.

СПИСОК ЛИТЕРАТУРЫ

1. Campbell Colin – Parallel Programming with Microsoft Visual C++, Microsoft corporation, 2011. – 172с.
2. М.П. Левин – Параллельное программирование с использованием OpenMP: учебное пособие/ М.П. Левин – М: Интернет университет информационных технологий; БИНОМ. Лаборатория знаний. 2012. – 118с.ил., табл. – (Серия «Основы информационных технологий»).
3. А.С. Антонов– Параллельное программирование с использованием технологии OpenMP: Учебное пособие – М.: Изд– во МГУ, 2009. – 77 с.
4. Библиотеки и документация MSNDMicrosoft, [Электронный ресурс]. URL: <http://msdn.microsoft.com/ru-ru/library /hh265137.aspx> (дата обращения 15.04.13)
5. ЭхтерШ.,Роберт Дж. – Многоядерное программирование. – СПб.: Питер,2010. – 316с.: ил– (Серия «Библиотека программиста»).
6. Dosi Marshall – Parallel Programming with Microsoft Visual Studio 2010, Microsoft, 2010. – 249с.
7. Barbara Chapman, Gabriele Jost, Ruud Van Der Pas – Using OpenMP, The

MIT Press, Cambridge, Massachusetts, 2008. – 378с.

8. Intel MIC, [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/Intel_MIC (дата обращения 15.02.16).

9. IntelXeonPhi, [Электронный ресурс]. URL: <http://www.intel.ru/content/www/ru/ru/processors/xeon/xeon-phi-detail.html> (дата обращения 15.02.16).

10. Обзор IntelXeonPhi, [Электронный ресурс]. URL: http://www.thg.ru/cpu/obzor_intel_xeon_phi/ (дата обращения 21.03.16).

11. Программирование для IntelXeonPhi, [Электронный ресурс]. URL: <http://itprojects.narfu.ru/grid/material2014/2014-GRID5-XeonPhi-Architecture.pdf> (дата обращения 28.03.16).

12. Д.К. Андрейченко, К.П. Андрейченко. – Моделирование, анализ и синтез комбинированных динамических систем: Учебное пособие. – Саратов: ООО «Издательский Дом «Райт- Экспо. – 2013 – 144с. – ISBN 978-5-4426-0018-6

13. Андрейченко Д.К., Андрейченко К.П. К теории устойчивости цилиндрического гидродинамического подвеса// Изв. РАН. Механика жидкости и газа. 2009. №1. С. 13-26.

14. Авторизация с помощью ключей и PuTTY, [Электронный ресурс]. URL: <https://habrahabr.ru/post/127521/> (дата обращения 27.03.16).

15. WinSCP, [Электронный ресурс]. URL: <https://ru.wikipedia.org/wiki/WinSCP> (дата обращения 3.04.16).

16. WinSCP Free SFTP, SCP and FTP client for Windows, [Электронный ресурс]. URL: <https://winscp.net/eng/docs/lang:ru> (дата обращения 3.04.16).

17. Права доступа и файловые флаги, [Электронный ресурс]. URL: <http://vds-admin.ru/freebsd/permissions-prava-dostupa-i-failovye-flagi-v-operatsionnykh-sistemakh-unix> (дата обращения 5.05.16).

18. x86_64 архитектура, [Электронный ресурс]. URL:

<https://ru.wikipedia.org/wiki/X86-64> (дата обращения 5.05.16).

19. Особенности кластеров, [Электронный ресурс]. URL: http://infolib.lotus.com/resources/portal/8.0.0/doc/ru/PT800ACD001/plan/plan_clusters_ovr.html (дата обращения 7.05.16).

20. Computercluster, [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Computer_cluster (дата обращения 7.05.16).