

«УТВЕРЖДАЮ»

Проректор по научной работе



Драгунов В.К.

« 16 » июня 2015 г.

Программа аспирантуры

Направление 09.06.01 Информатика и вычислительная техника

Направленность (специальность) 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

РАБОЧАЯ ПРОГРАММА

дисциплины по выбору

«Программное обеспечение высокопроизводительных систем»

Индекс дисциплины по учебному плану: Б1.В.ДВ.1.2

Всего: 108 часов

Семестр 1, в том числе

6 часов – контактная работа,
84 часа – самостоятельная работа,
18 часов – контроль

Программа составлена на основе федерального государственного образовательного стандарта высшего образования (уровень подготовки кадров высшей квалификации) по направлению подготовки 09.06.01 Информатика и вычислительная техника, утвержденного приказом Минобрнауки России от 30 июля 2014 г. № 875, и паспорта специальности, указанной в номенклатуре специальностей научных работников 05.13.11 Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей, утвержденной приказом Минобрнауки России от 25 февраля 2009 г. № 59.

ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью изучения дисциплины является освоение базовых архитектур современных высокопроизводительных систем и основных концепций создания программных приложений для них.

Задачами дисциплины являются:

- формирование знаний о современных высокопроизводительных архитектурах;
- изучение концепций и средств программного обеспечения для создания эффективных параллельных приложений;
- изучение встроенных механизмов *UNIX*-подобных операционных систем;
- освоение базовых принципов решения задач на параллельных и распределенных архитектурах;
- приобретение практических навыков создания эффективных параллельных приложений для рассмотренных архитектур.

В процессе освоения дисциплины **формируются следующие компетенции:**

- способность к критическому анализу и оценке современных научных достижений, генерированию новых идей при решении исследовательских и практических задач, в том числе в междисциплинарных областях (УК-1).

– готовность участвовать в работе российских и международных исследовательских коллективов по решению научных и научно-образовательных задач (УК-3).

– владение культурой научного исследования, в том числе с использованием современных информационно-коммуникационных технологий (ОПК-2).

– способность объективно оценивать результаты исследований и разработок, выполненных другими специалистами и в других научных учреждениях (ОПК-5).

– владение методами создания программ и программных систем для параллельной и распределенной обработки данных, языками и инструментальными средствами параллельного программирования (ПК-3).

– способность разрабатывать наукоемкое программное обеспечение, человеко-машинные интерфейсы и выполнять оценку качества, стандартизацию и сопровождение программных систем (ПК-7).

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБРАЗОВАНИЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ

В результате освоения дисциплины обучающийся должен демонстрировать **следующие результаты образования:**

знать:

– основные источники научно-технической информации (периодические издания, Интернет сайты), содержащие материалы по моделям обработки данных и архитектурам вычислительных систем (ОПК-2);

– классификацию и основные тенденции развития современных высокопроизводительных архитектур систем обработки данных (УК-1);

– основные классы моделей вычислений, их взаимосвязь с архитектурами вычислительных систем в указанной выше классификации (ПК-7);

– особенности функционирования операционных систем высокопроизводительных архитектур (ПК-3);

уметь:

– самостоятельно выбирать аппаратно-программные средства для решения поставленных задач (ОПК-5);

– разрабатывать и исследовать методы, алгоритмы и программные средства для решения сложных задач с помощью высокопроизводительных архитектур (ПК-3);

– осуществлять переход от модели обработки данных к архитектуре вычислительной системы, реализующей эту модель с различным уровнем качества (ПК-7);

– анализировать имеющуюся научно-техническую информацию, выбирать необходимые материалы, творчески развивать их (УК-3);

владеть:

– существующими инструментальными программными средствами для создания параллельных и распределенных программ (ПК-3);

– навыками поиска информации по моделям вычислений и архитектурам вычислительных систем (УК-1);

– навыками применения полученной информации при решении конкретных научно-технических задач (ОПК-5);

– терминологией, навыками дискуссии и публичных выступлений по рассматриваемой профессиональной тематике (УК-3);

– Методами анализа, верификации и тестирования программ и программных систем для высокопроизводительных архитектур (ПК-7).

КРАТКОЕ СОДЕРЖАНИЕ РАЗДЕЛОВ ДИСЦИПЛИНЫ

Основные концепции архитектур высокопроизводительных систем

(18 часов самостоятельной работы)

Основные тенденции развития современных высокопроизводительных средств вычислительной техники.

Влияние роста сложности задач на развитие архитектуры современного компьютера. Обзор классов сложных задач.

Классификации архитектур вычислительных систем. Примеры. Списки самых производительных систем TOP 500, TOP 50 и GREEN 500.

Конвейерные архитектуры. Суперскалярные процессоры. Векторные процессоры и векторная обработка данных. Параллельные компьютеры.

Связь между элементами параллельных вычислительных систем.

Понятие кластера. Кластеры рабочих станций. Особенности организации. Состав. Структуры узла. Современные коммуникационные технологии. Сравнительный анализ. Программное обеспечение, используемое для кластерных систем. Отечественные и зарубежные кластерные установки.

Графические процессоры (GPU). Параллельные вычисления и особенности программирования на GPU. Архитектура и программная модель CUDA.

Анализ возможностей и архитектура Интернет-технологий нового поколения (10 часов самостоятельной работы)

Общая концепция GRID-технологии. Архитектура GRID-сетей: базовые функции и базовые компоненты. Характеристики вычислительных ресурсов GRID-сетей. Коммерческое использование и реальные GRID-проекты.

Понятие метакомпьютинга и облачных вычислений. Характеристики облачных вычислений. Модели развертывания и модели обслуживания. Экономические аспекты и технологии.

Программное обеспечение современных высокопроизводительных систем (20 часов самостоятельной работы)

Обзор и классификация программного обеспечения для высокопроизводительных вычислительных систем. Средства создания программного обеспечения.

Основные концепции технологий программирования для систем с общей и распределенной памятью. Классификация современных параллельных технологий. Особенности применения.

OpenMP – технология программирования в модели общей памяти. Многопоточное программирование.

Системы программирования на основе передачи сообщений - PVM, MPI, MPI-2. Общие сведения и характерные особенности. Примеры использования. Интерфейсы для последовательных языков Fortran, C, C++. Примеры использования.

Программирование и использования аппаратных ускорителей для повышения производительности вычислений в высокопроизводительных системах. Система программирования Compute Unified Device Architecture (*CUDA*) для ускорителей на видеокартах *NVideo* и язык *OpenCL*.

Встроенные механизмы операционных систем

(12 часов самостоятельной работы)

Внутреннее устройство встроенных механизмов каналов, сигналов и разделяемой памяти *UNIX* и *UNIX*-подобных операционных систем, системные вызовы, функции и их использование. Механизмы семафоров и мониторов *UNIX* и *UNIX*-подобных операционных систем, а также системные вызовы, функции и их использование. Создание параллельных программ с использованием этих механизмов.

Механизмы очередей сообщений и сокетов *UNIX* и *UNIX*-подобных операционных систем, системные вызовы, функции и их использование. Файловые и сетевые сокеты, сетевое программирование. Использование этих механизмов при создании параллельных программ.

Основные принципы решения задач на параллельных и распределенных архитектурах (14 часов самостоятельной работы)

Основные принципы распараллеливания: глобальное распараллеливание (крупнозернистое, распараллеливание на уровне задач) на примере решения класса матричных задач и локальное (мелкозернистое) на примерах распараллеливания выражений и циклов.

Методы распараллеливания циклов.

Векторизация программ – как метод решения задач на векторных компьютерах. Способы векторизации программ. Оценка быстродействия векторных программ. Различие и сходство между распараллеливанием и векторизацией программ. Векторные операции и функции в языке программирования Фортран 90.

Методы декомпозиции – основа построения параллельных вычислительных алгоритмов (10 часов самостоятельной работы)

Декомпозиция для класса экстремальных задач большой размерности: декомпозиция на основе разделения переменных, параметрическая декомпозиция и др. Методы декомпозиции для задач линейной алгебры (блочные методы, методы декомпозиции области и их параллельные модификации).

Матричная и структурная декомпозиция для СЛАУ. Метод развертки матрицы коэффициентов и приведения к блочно-диагональному виду с окаймлениями. Параллельно – последовательные схемы вычислений. Оценки ускорения параллельного решения и эффективности.

ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ КОНТРОЛЯ ОСВОЕНИЯ РЕЗУЛЬТАТОВ ОБРАЗОВАНИЯ ПО ДИСЦИПЛИНЕ

Промежуточная аттестация по итогам освоения дисциплины:

1 семестр – дифференцированный зачет.

Вопросы для самоконтроля

1. Основные способы достижения параллелизма?
2. На чем основана классификация М. Флинна?
3. В чем состоит принцип разделения параллельных систем на мультипроцессоры и мультикомпьютеры?
4. Перечислите представителей класса мультипроцессоров
5. Определите положительные и отрицательные стороны симметричных мультипроцессоров?
6. Перечислите представителей класса мультикомпьютеров?
7. Определите положительные и отрицательные стороны кластерных систем?
8. Какие топологии сетей передачи данных наиболее широко используются при построении многопроцессорных систем?
9. В чем состоят особенности сетей передачи данных для кластеров?
10. Каковы основные характеристики сетей передачи данных?
11. Какие системные платформы могут быть использованы для построения кластеров?
12. Как определяется время выполнения параллельного алгоритма и как определить минимально возможное время решения задачи?
13. Что понимается под паракомпьютером и для чего может оказаться полезным данное понятие?
14. Какие оценки следует использовать в качестве характеристики времени последовательного решения задачи?
15. Какие зависимости могут быть получены для времени параллельного решения задачи при увеличении или уменьшении числа используемых процессоров?

16. При каком числе процессоров могут быть получены времена выполнения параллельного алгоритма, сопоставимые по порядку с оценками минимально возможного времени решения задачи?
17. Как определяются понятия ускорения и эффективности?
18. Возможно ли достижение сверхлинейного ускорения?
19. В чем состоит противоречивость показателей ускорения и эффективности?
20. Как определяется понятие стоимости вычислений?
21. В чем заключается проблема распараллеливания рекурсивных схем первого порядка?
22. Оценки ускорения и эффективности для вариантов реализации параллельных схем суммирования
23. Параллельный алгоритм вычисления всех частных сумм последовательности числовых значений
24. Как формулируется закон Амдаля? Какой аспект параллельных вычислений позволяет учесть данный закон?
25. Какие предположения используются для обоснования закона Густавсона-Барсиса?
26. Как определяется функция изоэффективности?
27. Какой алгоритм называется масштабируемым? Приведите примеры методов с разным уровнем масштабируемости.
28. Какие основные характеристики используются для оценки топологии сети передачи данных? Приведите значения характеристик для конкретных типов коммуникационных структур (полный граф, линейка, решетка, гиперкуб, и др.).
29. Какие основные методы применяются при маршрутизации передаваемых данных по сети?
30. В чем состоят основные методы передачи данных? Приведите для этих методов аналитические оценки времени выполнения.
31. Какие операции передачи данных могут быть выделены в качестве основных?

32. В чем состоят особенности алгоритмов передачи данных от одного процессора всем процессорам сети для топологии кольца, решетки и гиперкуба? Приведите оценки временной трудоемкости для этих алгоритмов?

33. В чем состоят особенности алгоритмов передачи данных от всех процессоров всем процессорам сети для топологии кольца, решетки и гиперкуба? Приведите оценки временной трудоемкости для этих алгоритмов.

34. В чем состоят особенности алгоритмов операции редукции? Какой из возможных алгоритмов является наилучшим по времени выполнения?

35. В чем состоят особенности алгоритма операции циклического сдвига?

36. Какой положительный эффект может дать механизм логических топологий? Приведите примеры алгоритмов логического представления структуры коммуникационной сети.

37. В чем состоит различие моделей для оценки времени выполнения операций передачи данных в кластерных вычислительных системах? Какая модель является более точной?

38. Какая модель может быть использована для предварительного анализа временной трудоемкости коммуникационных операций?

39. Каковы основные этапы проектирования и разработки методов параллельных вычислений?

40. Какие основные требования должны быть выполнены при разработке параллельных алгоритмов?

41. В чем состоят основные действия на этапе выделения подзадач?

42. Каковы основные действия на этапе определения информационных зависимостей?

43. В чем состоят действия на этапе масштабирования имеющегося набора подзадач?

44. В чем состоят основные действия на этапе распределения подзадач по процессорам вычислительной системы?

45. Как происходит динамическое управление распределением вычислительной нагрузки в модели «хозяин-исполнитель»?
46. Какой способ выполнения операции обобщенного сбора данных является более эффективным?
47. Какой минимальный набор средств является достаточным для организации параллельных вычислений в системах с распределенной памятью?
48. В чем состоит важность стандартизации средств передачи сообщений?
49. Что следует понимать под параллельной программой в модели с распределенной памятью?
50. В чем различие понятий процесса и процессора?
51. Какой минимальный набор функций MPI позволяет начать разработку параллельных программ?
52. Как описываются передаваемые MPI-сообщения?
53. Как можно организовать прием сообщений от конкретных процессов?
54. Как определить время выполнения MPI-программы?
55. В чем различие парных и коллективных операций передачи данных?
56. Какая функция MPI обеспечивает передачу данных от одного процесса всем процессам?
57. Что понимается под операцией редукции в MPI?
58. В каких ситуациях следует применять барьерную синхронизацию?
59. Какие режимы передачи данных поддерживаются в MPI?
60. Как организуется неблокирующий обмен данными в MPI?
61. В чем состоит понятие тупика? Когда функция одновременного выполнения передачи и приема гарантирует отсутствие тупиковых ситуаций?
62. Какие коллективные операции передачи данных предусмотрены в MPI?
63. Что понимается под производным типом данных в MPI?

64. Какие существуют способы конструирования производных типов данных в MPI?
65. В каких ситуациях может быть полезна упаковка и распаковка данных?
66. Что понимается в MPI под коммуникатором?
67. Для чего может потребоваться создание новых коммуникаторов?
68. Что понимается в MPI под виртуальной топологией?
69. Какие виды топологий предусмотрены в MPI?
70. Назовите основные способы распределения элементов двумерной матрицы между процессорами распределенной вычислительной системы.
71. Какова вычислительная сложность последовательного алгоритма умножения матрицы на вектор?
72. Почему при разработке параллельных алгоритмов умножения матрицы на вектор допустимо дублировать вектор-операнд на все процессоры?
73. Какие подходы могут быть предложены для разработки параллельных алгоритмов умножения матрицы на вектор?
74. Какие информационные взаимодействия выполняются для алгоритмов при ленточной схеме разделения данных? В чем различие необходимых операций передачи данных при разделении матрицы по строкам и столбцам?
75. Какие информационные взаимодействия выполняются для блочного алгоритма умножения матрицы на вектор?
76. Какие способы разделения данных целесообразно использовать при разработке параллельных алгоритмов матричного умножения?
77. Какие информационные взаимодействия выполняются для алгоритмов при ленточной схеме разделения данных? При блочной схеме?
78. Какова идея параллельной реализации метода Гаусса решения систем линейных уравнений?
79. Какие информационные взаимодействия имеются между базовыми подзадачами для параллельного варианта метода Гаусса?

80. В чем состоит задача поиска всех кратчайших путей в графе? Приведите общую схему алгоритма Флойда. Какова трудоемкость алгоритма?

81. Приведите общую схему алгоритма Прима построения минимального остовного дерева. Какова трудоемкость алгоритма? На чем основывается идея распараллеливания алгоритма Прима?

82. Чем различаются общие и локальные переменные в последовательной секции OpenMP-программы?

83. Допускает ли OpenMP изменение числа параллельных нитей в ходе работы программы?

Вопросы для проведения зачета

1. Классификации современных параллельных вычислительных систем (Флинна, Хокни, Фенга, Хендлера, Шнайдера, Скилликорна);

2. Мультипроцессоры и мультикомпьютеры. Топологии сетей передачи данных и их основные характеристики.

3. Способы повышения эффективности работы вычислительных систем: принципы конвейерной и параллельной обработки.

4. Показатели эффективности параллельного алгоритма, законы Амдаля и Густавсона-Барсиса. Оценка коммуникационной трудоемкости параллельных алгоритмов.

5. Концепция GRID и метакомпьютинг. Особенности распределения задач и передачи данных.

6. Технологии параллельного программирования. Модели программирования с общей и распределенной памятью

7. Технология MPI. Понятия сообщения, коммуникатора и программы в терминах MPI;

8. Технология MPI. Особенности парных и коллективных коммуникационных взаимодействий. Синхронные и асинхронные, блокирующие и неблокирующие обмены.

9. Технология MPI. Управление параллельной программой: понятие тупика и барьерная синхронизация. Производные типы данных, способы конструирования.

10. Технология OpenMP. Понятия процесса, нити и структурного блока; управляющие директивы OpenMP.

11. Технология OpenMP. Организация балансировки параллельных потоков; методы распараллеливания циклов, циклы с зависимостью по данным; редукционные операторы.

12. Технология OpenMP. Явление гонки потоков и основные конструкции для синхронизации нитей.

13. Средства для управления параллельной и распределенной обработкой. Управление параллельными процессами в ВС: архитектура, системы управления, планирование процессов и управление загруженностью.

14. Управление памятью. Распределение памяти разделами. Проблема фрагментации. Страничное, сегментное и сегментно–страничное распределение памяти. Управление виртуальной памятью. Стратегии замещения страниц.

15. Синхронизация процессов, семафоры, мониторы. Проблема тупиков: способы ее предотвращения и решения.

16. Иерархическая модель файловой системы, каталоги и дескрипторы файлов. Уровни файловой системы.

17. Сетевые ОС: структуры и варианты построения. Управления распределенными ресурсами.

18. Состояния потока. Вытесняющие и невытесняющие алгоритмы планирования. Алгоритмы планирования, основанные на квантовании. Алгоритмы планирования, основанные на приоритетах. Смешанные алгоритмы планирования.

19. Основные возможности операционной системы Unix. Особенности модели доступа доступны в системе Unix. Механизмы назначения прав доступа. Отличие жестких и символьных ссылок в файловой системе Unix.

20. Unix. Основные команды файловой системы.

Критерии оценки за освоение дисциплины определены в Инструктивном письме И-23 от 14 мая 2012 г.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

1. Таненбаум Э. Архитектура компьютера: пер. с англ. / – 5-е изд. – СПб.: Питер, 2011. – 844 с.
2. В.П. Гергель. Современные языки и технологии параллельного программирования: учебник для вузов. Б-ка Нижегородского гос. ун-та им. Н.И. Лобачевского. – М.: Изд-во Моск. ун-та, 2012. – 408 с.
3. С.А. Лупин, В.Д. Колдаев. Архитектура ЭВМ: учебное пособие. – М.: Форум: ИНФРА-М, 2011. – 384 с.
4. Робачевский А., Немнюгин С., Стесик О. Операционная система Unix. – 2-е изд. – СПб.: БХВ-Петербург, 2012. – 636 с.
5. Левин М.П. Параллельное программирование с использованием OpenMP: учебное пособие. – М.: Интернет-Ун-т информ. технологий: БИНОМ. Лаборатория знаний, 2012. – 118 с.

Дополнительная литература:

6. В.В. Воеводин, Вл.В. Воеводин. Параллельные вычисления: Научное издание: Учебное пособие для вузов. – СПб.: БХВ-Петербург, 2002. – 608 с.
7. И.И. Ладыгин, А.В. Логинов, А.В. Филатов, С.Г. Яньков. Кластеры на многоядерных процессорах: учебное пособие. – М.: Изд. дом МЭИ, 2008. – 112 с.
8. Суперкомпьютерные технологии в науке, образовании и промышленности / В.А. Садовничий и др., Моск. гос. ун-т им. М.В. Ломоносова (МГУ), Суперкомпьютерный консорциум ун-тов России. РАН; Ред. В.А. Садовничий, и др. – М.: Изд-во Моск. ун-та, 2012. – 232 с.
9. Малышкин В.Э., Корнеев В.Д. Параллельное программирование мультикомпьютеров. – Новосибирск: Изд-во НГТУ, 2011. – 296 с.

10. Боресков А.В. и др. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: учебное пособие для вузов, Моск. гос. ун-т им. М.В. Ломоносова (МГУ). – М.: Изд-во Моск. ун-та, 2012. – 336 с.