**Федеральное государственное автономное образовательное учреждение высшего профессионального образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Московский институт электроники и математики

Департамент прикладной математики

**Рабочая программа дисциплины**
«Атомистическое моделирование и суперкомпьютеры»

для образовательной программы «Математические методы моделирования и

компьютерные технологии» направления подготовки 01.04.02 «Прикладная математика и информатика» магистра

Разработчик программы: И.В. Морозов, к. ф.-м. н., доцент, morozov@ihed.ras.ru

Одобрена на заседании департамента прикладной математики

 «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г.

Руководитель департамента А. В. Белов \_\_\_\_\_\_\_\_ [подпись]

Рекомендована Академическим советом образовательной программы

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г., № протокола\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждена «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2015 г.

Академический руководитель образовательной программы

М.В. Карасев \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ [подпись]

Москва, 2015

*Настоящая программа не может быть использована другими подразделениями университета и другими вузами без разрешения подразделения-разработчика программы.*

# Область применения и нормативные ссылки

Настоящая программа учебной дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Программа предназначена для преподавателей, ведущих данную дисциплину, учебных ассистентов и студентов направления подготовки **01.04.02 «Прикладная математика и информатики»**, обучающихся по магистерской программе **«Математические методы моделирования и компьютерные технологии»,** изучающих дисциплину «**Атомистическое моделирование и суперкомпьютеры**».

Программа разработана в соответствии с:

* Образовательным стандартом государственного образовательного бюджетного учреждения высшего профессионального образования «Государственный университет – Высшая школа экономики», в отношении которого установлена категория «Национальный исследовательский университет»;
* Образовательной программой «Математические методы моделирования и компьютерные технологии» для направления 01.04.02 «Прикладная математика и информатика» подготовки магистра;
* Рабочим учебным планом университета по направлению 01.04.02 «Прикладная математика и информатика» подготовки магистра по программе «Математические методы моделирования и компьютерные технологии», утвержденным в 2015г.

# Цели освоения дисциплины

Целями освоения дисциплины «Атомистическое моделирование и суперкомпьютеры» является ознакомление студентов с методами атомистического моделирования в физике, химии и биологии, а также получение навыка использования современных суперкомпьютеров для проведения численных экспериментов с применением указанных методов.

В результате выполнения заданий по курсу студенты приобретают навыки:

* постановки численного эксперимента с использованием методов атомистического моделирования для прикладных и фундаментальных исследований в естественных науках;
* создания программ молекулярно-динамического моделирования, а также использования готовых пакетов программ;
* работы на суперкомпьютерных вычислительных системах в качестве пользователя;
* разработки параллельных программ для систем с общей и распределенной памятью.

# Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины студент должен:

* *знать* область применения современных суперкомпьютеров, роль компьютерного моделирования в современных научных исследованиях;
* *знать* сферу применения методов атомистического моделирования в задачах физики, химии, биологии;
* *знать* основные законы и формулы, необходимые для построения численных схем, граничных и начальных условий, моделей взаимодействия частиц в методах молекулярной динамики и Монте-Карло;
* *знать* архитектуру и основные характеристики современных суперкомпьютерных систем;
* *уметь* проектировать и создавать новые параллельные программы, выбирать оптимальные алгоритмы распараллеливания, в том числе, для задач атомистического моделирования;
* *уметь* анализировать результаты атомистического моделирования и обобщать полученные данные;
* *уметь* компилировать и запускать программы на суперкомпьютерных кластерах, контролировать правильность их выполнения, выявлять и исправлять типичные ошибки;
* *иметь навыки* работы со стандартным программным обеспечением суперкомпьютерных кластеров;
* *иметь навыки* работы с наиболее распространенными пакетами атомистического моделирования;
* *иметь навыки* создания и отладки параллельных программ на суперкомпьютерных кластерах;
* *иметь навыки* проведения простейших численных экспериментов методам молекулярной динамики и Монте-Карло.

В результате освоения дисциплины студент осваивает следующие компетенции:

| Компетенция | Код по ФГОС/ НИУ | Дескрипторы – основные признаки освоения (показатели достижения результата) | Формы и методы обучения, способствующие формированию и развитию компетенции |
| --- | --- | --- | --- |
| Способен рефлексировать (оценивать и перерабатывать) освоенные научные методы и способы деятельности.  | СК-М1  | Демонстрирует понимание принципов построения численных моделей и стандартных алгоритмов в методах молекулярной динамики и Монте-Карло | Лекции, самостоятельная работа |
| Способен анализировать, верифицировать, оценивать полноту информации в ходе профессиональной деятельности, при необходимости восполнять и синтезировать недостающую информацию | СК-М6  | Использует знания о принципах моделирования для создания собственных алгоритмов при решении конкретных задач физики, химии и биологии | Практические занятия, выполнение домашних работ |
| Способен анализировать и воспроизводить смысл междисциплинарных текстов с использованием языка и аппарата прикладной математики | ИК-М2.1пми  | Использует аппарат статистической физики, вычислительной математики и информатики для постановки численного эксперимента, создания программ моделирования и анализа результатов | Лекции, практические занятия и самостоятельная работа |
| Способен строить и решать математические модели в соответствии с направлением подготовки и специализацией | ИК-М7.2пми  | Использует данные моделирования для более глубокого понимания процессов на микро- и нано- уровнях описания веществ | Лекции, практические занятия и самостоятельная работа |
| Способен понимать и применять в исследовательской и прикладной деятельности современный математический аппарат  | ИК-М7.3пми  | Использует методы атомистического моделирования для решения исследовательских и прикладных задач физики, химии, биологии | Практические занятия, выполнение домашних работ |

# Место дисциплины в структуре образовательной программы

Настоящая дисциплина относится к базовой части цикла дисциплин программы/специализации.

Для освоения учебной дисциплины, студенты должны владеть следующими знаниями и компетенциями:

* навыками работы с компьютером в качестве пользователя;
* знанием и практическими навыками разработки программ на алгоритмических языках C или Fortran;
* знаниями основ математического анализа;
* знанием основных положений курса общей физики;
* навыками решения типовых задач курса «Вычислительная математика».

Основные положения дисциплины должны быть использованы в дальнейшем при изучении следующих дисциплин:

* Многомасштабное компьютерное моделирование;
* Компьютерная молекулярная биология и медицина;
* прочие курсы, программа которых, затрагивает вопросы проведения масштабных численных экспериментов.

# Тематический план учебной дисциплины

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Название раздела | Всего часов  | Аудиторные часы | Самостоя­тельная работа |
| Лекции | Семинары | Практические занятия |
| 1 | Архитектура и принципы работы суперкомпьютеров | 20 | 4 |  | 0 | 16 |
| 2 | Параллельное программирование для систем с общей памятью | 58 | 8 |  | 12 | 38 |
| 3 | Параллельное программирование для систем с распределенной памятью | 48 | 6 |  | 10 | 32 |
| 4 | Методы атомистического моделирования | 62 | 8 |  | 16 | 38 |
| 5 | Оптимизация и распараллеливание в задачах атомистического моделирования | 40 | 4 |  | 8 | 28 |
|  | **Всего** | **228** | **30** |  | **46** |  **152** |

# Формы контроля знаний студентов

|  |  |  |  |
| --- | --- | --- | --- |
| Тип контроля | Форма контроля | 1 год | Параметры  |
| 3 | 4 |
| Итоговый | Экзамен |  | Х | Устный экзамен на 120 минут. Включает от 3 до 5 вопросов по всем темам курса различного уровня сложности. |

## Критерии оценки знаний, навыков

Оценки по всем формам текущего контроля выставляются по 10-ти балльной шкале.

# Содержание дисциплины

Содержание дисциплины разбито на разделы, каждый включает в себя 4 темы по которым проводится одна или две лекции и один или два практических занятия.

1. Раздел 1. Архитектура и принципы работы суперкомпьютеров.

Тема 1. Обзор высокопроизводительных систем в России и за рубежом. Обсуждение последних редакций рейтингов Top-500 и Top-50. Качественный переход от последовательных к массивно-параллельным архитектурам и алгоритмам. Технологические проблемы повышения быстродействия компьютеров, путь к экзафлопсной производительности. Проблемы энергопотребления и надежности суперкомпьютеров. История и направление развития методов атомистического моделирования, необходимость применения суперкомпьютеров. (лекции – 2 часа)

Тема 2. Классификация вычислительных систем. Параллелизм по задачам и по данным. Системы с общей и распределенной памятью. Внутренний параллелизм современных процессоров, скалярная и суперскалярная архитекутры, конвейер команд. Многоядерные процессоры. Модели взаимодействия с памятью UMA и NUMA. Перспективы наращивания числа ядер, проблема когерентности кэша. (лекции – 2 часа)

Литература по разделу:

1. Гергель В.П. Теория и практика параллельных вычислений. Учебное пособие. М: Бином, 2007.
2. Суперкомпьютерные технологии в науке, образовании и промышленности / Под ред.: В.А. Садовничего, академика Г.И. Савина, Вл.В. Воеводина. М.: МГУ, 2009.
3. Сайт Лаборатории параллельных информационных технологий НИВЦ МГУ http://parallel.ru
4. Раздел 2. Параллельное программирование для систем с общей памятью.

Тема 1. Особенности создания параллельных программ для систем с общей памятью. Поддержка параллелизма на уровне операционной системы. Процессы (process) и потоки (threads). Создание многопоточных программ с использованием базовых средств операционных систем. (лекции – 2 часа, практ. – 4 часа)

Тема 2. Синхронизация потоков и детерминированность результатов работы программы. Локальные и общие переменные потоков, безопасный доступ к общим переменным. Побочные эффекты, реентерабельность процедур. Избыточная синхронизации потоков, тупики. (лекции – 2 часа, практ. – 4 часа)

Тема 3. Распараллеливание программ с использованием технологии OpenMP. Использование высокоуровневых библиотек и параллельных языков программирования. Автоматическое распараллеливанием программ. Отладка параллельных программ. (лекции – 2 часа, практ. – 4 часа)

Тема 4. Теоретические основы параллельных алгоритмов. Понятия загруженности, производительности и ускорения. Информационная зависимость операций, графы исполнения. Достаточные условия Бернстайна. Распараллеливание циклов с информационными зависимостями. (лекции – 2 часа)

Литература по разделу:

1. Демьянович, Ю. К. Параллельные алгоритмы. Разработка и реализация. М. БИНОМ. Лаборатория знаний, 2012. – 343 с.
2. Карпов В.Е., Коньков К.А. Основы операционных систем. М.: Интуит, 2004.
3. Butenhof D.R. Programming with POSIX Threads. Addison-Wesley, 1997.
4. Левин М.П. Параллельное программирование с использованием OpenMP. М: Интуит, 2012.
5. Антонов А.С. Параллельное программирование с использованием технологии OpenMP. М.: МГУ, 2009
6. Карпов В.Е., Лобанов А.И. Численные методы, алгоритмы и программы. Введение в распараллеливание. М.: Физматкнига, 2014. – 196 с.
7. Официальная документация по OpenMP: http://www.openmp.org (на англ. языке)
8. Введение в OpenMP: <https://software.intel.com/ru-ru/blogs/2011/11/21/openmp-c?language=ru>
9. Учебное пособие по OpenMP <http://www.llnl.gov/computing/tutorials/openMP> (на англ. языке)
10. Раздел 3. Параллельное программирование для систем с распределенной памятью.

Тема 1. Программное обеспечение суперкомпьютерного кластера. Кластеры типа Beowulf. Использование системы управления очередями задач (PBS, SLURM). (лекции – 2 часа, практ. – 2 часа)

Тема 2. Технология MPI. Классификация функций MPI и основные понятия. Компиляция и запуск программ. Функции двухточечного обмена сообщениями. Функции коллективного обмена сообщениями. (лекции – 2 часа, практ. – 4 часа)

Тема 3. Односторонняя и двухсторонняя модели обмена сообщениями. Дополнительные возможности стандарта MPI-2. (лекции – 2 часа, практ. – 4 часа)

Литература по разделу:

1. Антонов А.С. Александр Антонов: Технологии параллельного программирования MPI и OpenMP. Учебное пособие. М.: МГУ, 2013
2. Корнеев В.Д. Параллельное программирование в MPI. Новосибирск: Изд-во ИВМиМГ СО РАН, 2002.
3. Шпаковский Г. И., Верхотуров А. Е., Серикова Н. В. Руководство по работе на вычислительном кластере. Учебное пособие. Минск.: БГУ, 2004.
4. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. М.: БХВ-Санкт-Петербург, 2004.
5. Богачёв К.Ю. Основы параллельного программирования, М: Бином, 2003.
6. D.H.M. Spector. Building Linux Clusters. O'Reilly Media, 2000.
7. Официальная документация по стандартам MPI: <http://www.mcs.anl.gov/mpi> (на англ. языке)
8. Евсеев И. MPI для начинающих: <http://www2.sscc.ru/Links/Litera/il/default.htm>
9. Раздел 4. Методы атомистического моделирования.

Тема 1. Основы метода молекулярной динамики. Решение уравнений движения частиц. Ошибки интегрирования и ошибки округления. Точность сохранения энергии в МД системе. Выбор оптимального шага по времени. (лекции – 2 часа, практ. – 4 часа)

Тема 2. Начальные и граничные условия при интегрировании уравнений движения. Метод ближайшего образа. Применение термостатов и баростатов. (лекции – 2 часа, практ. – 4 часа)

Тема 3. Иерархия потенциалов взаимодействия частиц для различной степени детализации моделируемой системы. Модели взаимодействия нейтральных атомов и молекул, силовые поля для биологических систем, многочастичные потенциалы для металлов. Явное моделирование динамики электронов, классическая и квантовая молекулярная динамика. (лекции – 2 часа, практ. – 4 часа)

Тема 4. Метод Монте-Карло для моделирования систем многих частиц. История и обоснование метода. Алгоритм Метрополиса. Выбор амплитуды случайных источников. Обзор пакетов молекулярно-динамического моделирования. (лекции – 2 часа, практ. – 4 часа)

Литература по разделу:

1. Смит Б., Френкель Д. Принципы компьютерного моделирования молекулярных систем: от алгоритмов к приложениям. Научный мир, 2013. – 578 с.
2. Рапапорт Д. К. Искусство молекулярной динамики. М.: РХД НИЦ, 2012.
3. Allen M.P., Tildesley D.J. Computer Simulation of Liquids. Oxford : Clarendon Press, 1989. – 285 с.
4. Зленко Д.В., Мамонов П.А., Нестеренко А.М. Современные методы молекулярного моделирования. М.: МГУ, 2007.
5. Х. Гулд, Я. Тобочник. Компьютерное моделирование в физике, М.: Наука, 1990. Т. 2. – 400 с.
6. Кривцов А.М., Кривцова Н.В., Метод частиц и его использование в механике деформируемого твердого тела // Дальневосточный математический журнал ДВО РАН. 2002. Т. 3. № 2. C. 254-276.
7. Sutmann G., Classical molecular dynamics // In: Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms (eds. J. Grotendorst, et al). Julich: NIC. 2002. V. 10. P. 211-254, 2002.
8. Бэдсел Ч., Ленгдон А. Физика плазмы и численное моделирование. М.: Энергоатомиздат, 1989. – 452 с.
9. Kuksin A.Yu., Morozov I.V., Norman G.E., Stegailov V.V., Valuev I.A. Standards for Molecular Dynamics Modelling and Simulation of Relaxation // Molecular Simulation. 2005. V. 31. № 14 –15. P. 1005-1017.
10. Официальная страница проекта LAMMPS: <http://lammps.sandia.gov> (на англ. языке)
11. Раздел 5. Оптимизация и распараллеливание в задачах атомистического моделирования.

Тема 1. Оптимизация расчета взаимодействия частиц, списки Верле, связанные списки частиц в ячейках. Параллельные алгоритмы: декомпозиция по частицам и по пространству. Эффективность распараллеливания. Оптимизация для дальнодействующих потенциалов. (лекции – 2 часа, практ. – 4 часа)

Тема 2. Применение графических ускорителей (ГУ) для вычислений, не связанных с обработкой графических изображений. Архитектура ГУ, организация памяти и избежание задержек, связанных с обращением к памяти. Средства разработки программ для ГУ. Кластеры на основе гибридных систем, включающих ГУ. Эффективность применения ГУ для задач атомистического моделирования. (лекции – 2 часа, практ. – 4 часа)

Литература по разделу:

1. Verlet L. Computer "Experiments" on Classical Fluids. Phys. Rev., v. 159, pp. 98-103, 1967; v. 165, pp. 201-214, 1968; Phys. Rev A., v. 2, pp. 2514-2528, 1970; v. 7, pp. 1690-1700, 1973.
2. Gibbon P., Sutmann G. Long-Range Interactions in Many-Particle Simulation // In: Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms (eds. J. Grotendorst, et al). Julich: NIC. 2002. V. 10. P. 467-506.
3. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров. М.: ДМК Пресс, 2011.
4. Боресков А. В., Харламов А. А. Основы работы с технологией CUDA. М.: ДМК Пресс, 2012.
5. А. В. Боресков и др. Предисл.: В. А. Садовничий. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учеб. пособие. М.: МГУ, 2012.
6. Официальный сайт NVIDIA CUDA <http://www.nvidia.ru/object/cuda_home_new_ru.html>

Самостоятельная работа в количестве 64 часов включает в себя выполнение домашнего задания (30 часов), подготовку к практическим занятиям (18 часов) и подготовку к экзамену (16 часов).

# Образовательные технологии

Лекции по курсу сопровождаются презентациями с применением проектора, а также демонстрацией выполнения программ компьютерного моделирования на учебном суперкомпьютерном кластере.

## Методические рекомендации преподавателю

Дисциплина «Атомистическое моделирование и суперкомпьютеры», во многом продолжает знакомство студентов с компьютерным моделированием, основываясь на знаниях, полученных студентами ранее на курсах основ информатики и вычислительной математики. Курс с одной стороны дополняет эти знания, знакомя студентов с методами параллельного программирования, а с другой стороны, помогает осуществить переход от накопления теоретических знаний к их практическому применению. Это происходит за счет знакомства с актуальными проблемами вычислительной физики на примере задач атомистического моделирования.

Лекции по курсу сопровождаются практическими занятиями, на которых студенты приобретают навыки параллельного программирования, а также решения простейших задач методами молекулярной динамики и Монте-Карло.

Успешное освоение дисциплины требует регулярной самостоятельной работы студента. Самостоятельная работа включает в себя создание и отладку программ моделирования в рамках набора задач, рассматриваемых на практикумах и в рамках домашнего задания. Руководство и контроль за самостоятельной работой студента осуществляется преподавателем в форме обсуждения и сдачи индивидуальных задач на практических занятиях.

## Методические указания студентам

Особое место в самостоятельной работе студента занимает подготовка к практическим занятиям и выполнение домашних заданий. Это дает возможность приобрести необходимые практические навыки и способствует боле адекватной оценке работы студента со стороны преподавателя.

# Оценочные средства для текущего контроля и аттестации студента

## Тематика заданий текущего контроля

Ниже приведены примеры заданий для выполнения на практических занятиях и в форме домашнего задания.

1. Синхронизация потоков в системах с общей памятью. Добавить функции синхронизации в предложенную преподавателем программу, чтобы исключить потенциальную возможность конкуренции потоков («race condition»). Скомпилировать и проверить правильность работы полученной программы.
2. Блочное распараллеливание на POSIX Threads. Написать программу вычисления определенного интеграла методом трапеций с блочным распараллеливанием, используя POSIX Threads. Убедиться в правильности вычисления интеграла для произвольного числа потоков. Используя утилиту time построить зависимость ускорения от числа потоков. Для уменьшения погрешности результата время работы последовательной программы должно составлять не менее 30 сек.
3. Блочное распараллеливание на OpenMP. Написать программу вычисления определенного интеграла методом трапеций с блочным распараллеливанием, используя OpenMP. Убедиться в правильности вычисления интеграла для произвольного числа потоков. Используя утилиту time построить зависимость ускорения от числа потоков. Для уменьшения погрешности результата время работы последовательной программы должно составлять не менее 30 сек.
4. Блочно-циклическое распараллеливание на OpenMP. Написать программу численного интегрирования, в которой сложность расчета подынтегральной функции зависит от аргумента (пример: f(x) = x\*exp(x) при a < x < x0 и f(x) = x\*exp(sin(x)\*cos(x))/(2+cos(x\*x)) при x0 < x < b). Реализовать схему блочно-циклического распараллеливания на OpenMP. Построить зависимость ускорения от числа потоков для минимального, промежуточного и максимального размера блока.
5. Блочное распараллеливание на MPI. Вычисление определенного интеграла с блочным распараллеливанием на MPI. Необходимо подобрать функцию и число шагов так, чтобы характерное время работы последовательного варианта программы (один процесс) было около 1 минуты. В качестве результата требуется построить график зависимости времени выполнения программы и эффективности распараллеливания от числа процессов в интервале 1-16 процессов.
6. Блочно-циклическое распараллеливание на MPI. Вычисление определенного интеграла с блочно-циклическим распараллеливанием на MPI. Необходимо подобрать функцию и число шагов так, чтобы характерное время работы последовательного варианта программы (один процесс) было около 1 минуты. В качестве результата требуется построить график зависимости времени выполнения программы и эффективности распараллеливания от числа процессов в интервале 1-16 процессов.
7. Индивидуальное задание по методу молекулярной динамики или Монте-Карло. Написать программу моделирования Леннард-Джонсоновской жидкости с периодическими граничными условиями (методом ближайшего образа). Вывести системы на равновесие при заданной температуре и плотности, рассчитать термодинамические параметры и равновесные функции распределения (по выбору преподавателя). Проверить правильность работы программы, сравнив результат с имеющимися в литературе данными. Определить скорость работы последовательного алгоритма в зависимости от размера системы. Распараллелить программу с использованием OpenMP или MPI (по согласованию с преподавателем). Исследовать эффективность распараллеливания, построив зависимость ускорения от числа процессов (потоков исполнения).
8. Работа с пакетом молекулярно-динамического моделирования LAMMPS. Изучение формата входных параметров, запуск последовательной и параллельной версий, проведение тестового моделирования леннард-джонсоновской жидкости, исследование быстродействия, эффективности распараллеливания, ускорения на ГУ, обработка выходных данных.

## Вопросы для оценки качества освоения дисциплины

1. Классификация вычислительных систем, таксономия Флинна. Примеры систем различного типа.
2. Параллельные алгоритмы: распараллеливание по задачам и по данным.
3. Параллельные системы с общей памятью (SMP): ограничение на количество процессоров. Архитектуры UMA и NUMA.
4. Потоки (threads) и процессы (process) в многозадачных операционных системах. Выделение памяти, процессорного времени и других ресурсов процессам и потокам.
5. Создание потоков с использованием API POSIX Threads.
6. Синхронизация потоков. Примеры ошибок, связанных с отсутствием синхронизации. Объекты синхронизации POSIX Threads: взаимное исключение, сигнал, объект condvar.
7. Синхронизация потоков. Тупики (deadlocks) и необходимые условия их возникновения, принципы обнаружения и устранения тупиков.
8. Принцип программирования и компиляция программ с использованием OpenMP. Общие и локальные переменные потоков. Директивы распараллеливания.
9. Блочное и циклическое распараллеливание циклов. Алгоритмы распределения работы по потокам в OpenMP, директивы omp for и omp task.
10. Системы с распределенной памятью (MPP): особенности программирования по сравнению с SMP системами.
11. Общая схема программы с использованием библиотеки MPI. Компиляция и запуск MPI-программ.
12. Запуск последовательных и MPI-программ с использованием системы очередей PBS.
13. Функции передачи сообщений между двумя процессами в MPI. Классификация функций по способу синхронизации.
14. Блокирующие и неблокирующие функции приема-передачи в MPI.
15. Способы передача разнородных данных в одном сообщении в MPI.
16. Функции коллективного обмена сообщениями в MPI.
17. Односторонние коммуникации в MPI-2.
18. Использование графических ускорителей (ГУ) для научных вычислений. Особенности архитектуры ГУ. За счет чего достигается ускорение по сравнению с обычными процессорами? Основные проблемы и ограничения при написании программ для ГУ.
19. Численное интегрирование уравнений движения частиц в молекулярно-динамической (МД) системе c применением разностных схем Эйлера и Верле (Leap-Frog). Требования к потенциалу взаимодействия.
20. Выбор шага интегрирования и оптимальной разностной схемы. Точность сохранения полной энергии и импульса при моделировании NVE ансамбля.
21. Основные типы потенциалов взаимодействия для жидкостей и конденсированного состояния. Потенциал Леннарда-Джонса.
22. Граничные условия для МД ячейки. Метод ближайшего образа. Критерии выбора числа частиц.
23. Схема МД эксперимента. Как задать начальное состояние системы? Вывод МД системы на равновесие, использование терпостатов.
24. Метод Монте-Карло (МК) для моделирования систем многих частиц. Алгоритм Метрополиса.
25. Распараллеливание расчета взаимодействий между частицами: декомпозиция по пространству и по частицам.

## Примеры заданий промежуточного /итогового контроля

В дополнение к вопросам, приведенным в п. 9.2, билеты для итогового экзамена могут содержать следующие задачи:

* 1. Написать параллельную версию цикла с использованием директив OpenMP:

for(i=0; i<n-1; ++i) {

 a[i] = f(i);

 b[i] = a[i+1]\*a[i+1];

}

* 1. Написать параллельную версию цикла с использованием директив OpenMP:

for(i=1; i<n; ++i) {

 a[i] = f(i);

 b[i] = a[i-1]\*b[i];

}

* 1. Написать параллельную версию цикла с использованием директив OpenMP:

for(i=4; i<n; ++i)

 a[i] = f(a[i-4]);

* 1. Написать параллельную версию цикла с использованием директив OpenMP:

s = 0;

for(i=0; i<n-1; ++i)

 s += a[i]\*a[i+1];

* 1. Написать параллельную версию цикла с использованием директив OpenMP:

for(i=0; i<n-1; ++i)

 a[i] = f(a[i+1]);

* 1. MPI-программа содержит следующие определения и вызовы функций:

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

double data = (rank == 0) ? 1.234 : 0;

Написать фрагмент программы, в которой когда каждый процесс с rank>0 ожидает данные (data) от процесса rank-1, а затем передает их процессу rank+1 (алгоритм «эстафета»).

* 1. MPI-программа содержит следующие определения и вызовы функций:

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

double data = rank\*rank;

Написать фрагмент программы, в которой каждый процесс с нечетным номером rank обменивается данными data процессом с номером rank-1 (считается, что всего запущено четное число процессов).

* 1. MPI-программа содержит следующие определения и вызовы функций:

MPI\_Comm\_size(MPI\_COMM\_WORLD, &size);

MPI\_Comm\_rank(MPI\_COMM\_WORLD, &rank);

double data = rank\*rank;

Написать фрагмент программы, в которой каждый процесс с нечетным номером rank передает данные data процессу с номером rank-1 (считается, что всего запущено четное число процессов).

# Порядок формирования оценок по дисциплине

Преподаватель оценивает работу студентов на практических занятиях (Оауд), основываясь на результатах выполнения ими основных и дополнительных практических заданий, перечисленных в п. 9.1. Успешное выполнение одного задания оценивается в два балла. Также преподаватель может учесть активную работу студента на лекционных занятиях.

Домашняя работа студентов состоит в выполнении индивидуального задания по разработке программы моделирования, которое каждый из студентов получается в середине семестра. При защите задания на практическом занятии студент должен представить текст программы, продемонстрировать ее работоспособность и предъявить полученные с ее помощью результаты в виде таблиц и графиков. При этом студент должен ответить на вопросы преподавателя о назначении тех или иных блоков кода в тексте программы и объяснить смысл полученных результатов. По результатам этой работы оценивается самостоятельная работа студента (Осам.работа).

Суммарная накопленная оценка учитывает результаты студента следующим образом:

О*накопленная* = 0,5 \* Оауд + 0,5 \* Осам.работа

Способ округления оценки накопленной оценки: в пользу студента.

Итоговая оценка по 10-балльной шкале формируется как взвешенная сумма и учитывает результаты студента в течение семестра (О*накопленная*), а также результаты итогового экзамена, оценка за который выставляет как доля правильных ответов на вопросы, умноженная на 10 (О*итог.экз.*):

|  |
| --- |
| О*промежуточная* = 0,6 \* О*накопленная +* 0,4 \* О*итог.экз.* |

Способ округления оценки промежуточной аттестации: в пользу студента.

На экзамене студент может получить дополнительный вопрос (дополнительную практическую задачу), ответ на который оценивается в 1 балл.

# Учебно-методическое и информационное обеспечение дисциплины

## Базовый учебник

Нет

## Основная литература

1. Демьянович, Ю. К. Параллельные алгоритмы. Разработка и реализация. М. БИНОМ. Лаборатория знаний, 2012. – 343 с.
2. Антонов А.С. Александр Антонов: Технологии параллельного программирования MPI и OpenMP. Учебное пособие. М.: МГУ, 2013
3. Карпов В.Е., Лобанов А.И. Численные методы, алгоритмы и программы. Введение в распараллеливание. М.: Физматкнига, 2014. – 196 с.
4. Смит Б., Френкель Д. Принципы компьютерного моделирования молекулярных систем: от алгоритмов к приложениям. Научный мир, 2013. – 578 с.
5. Рапапорт Д. К. Искусство молекулярной динамики. М.: РХД НИЦ, 2012.

## Дополнительная литература

1. Суперкомпьютерные технологии в науке, образовании и промышленности / Под ред.: В.А. Садовничего, академика Г.И. Савина, Вл.В. Воеводина. М.: МГУ, 2009.
2. Карпов В.Е., Коньков К.А. Основы операционных систем. М.: Интуит, 2004.
3. Butenhof D.R. Programming with POSIX Threads. Addison-Wesley, 1997.
4. Левин М.П. Параллельное программирование с использованием OpenMP. М: Интуит, 2012.
5. Антонов А.С. Параллельное программирование с использованием технологии OpenMP. М.: МГУ, 2009
6. Корнеев В.Д. Параллельное программирование в MPI. Новосибирск: Изд-во ИВМиМГ СО РАН, 2002.
7. Шпаковский Г. И., Верхотуров А. Е., Серикова Н. В. Руководство по работе на вычислительном кластере. Учебное пособие. Минск.: БГУ, 2004.
8. Богачёв К.Ю. Основы параллельного программирования, М: Бином, 2003.
9. Гергель В.П. Теория и практика параллельных вычислений. Учебное пособие. М: Бином, 2007.
10. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. М.: БХВ-Санкт-Петербург, 2004.
11. D.H.M. Spector. Building Linux Clusters. O'Reilly Media, 2000.
12. Зленко Д.В., Мамонов П.А., Нестеренко А.М. Современные методы молекулярного моделирования. М.: МГУ, 2007.
13. Х. Гулд, Я. Тобочник. Компьютерное моделирование в физике, М.: Наука, 1990. Т. 2. – 400 с.
14. Кривцов А.М., Кривцова Н.В., Метод частиц и его использование в механике деформируемого твердого тела // Дальневосточный математический журнал ДВО РАН. 2002. Т. 3. № 2. C. 254-276.
15. Allen M.P., Tildesley D.J. Computer Simulation of Liquids. Oxford : Clarendon Press, 1989. – 285 с.
16. Sutmann G., Classical molecular dynamics // In: Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms (eds. J. Grotendorst, et al). Julich: NIC. 2002. V. 10. P. 211-254, 2002.
17. Бэдсел Ч., Ленгдон А. Физика плазмы и численное моделирование. М.: Энергоатомиздат, 1989. – 452 с.
18. Kuksin A.Yu., Morozov I.V., Norman G.E., Stegailov V.V., Valuev I.A. Standards for Molecular Dynamics Modelling and Simulation of Relaxation // Molecular Simulation. 2005. V. 31. № 14 –15. P. 1005-1017.
19. Verlet L. Computer "Experiments" on Classical Fluids. Phys. Rev., v. 159, pp. 98-103, 1967; v. 165, pp. 201-214, 1968; Phys. Rev A., v. 2, pp. 2514-2528, 1970; v. 7, pp. 1690-1700, 1973.
20. Gibbon P., Sutmann G. Long-Range Interactions in Many-Particle Simulation // In: Quantum Simulations of Complex Many-Body Systems: From Theory to Algorithms (eds. J. Grotendorst, et al). Julich: NIC. 2002. V. 10. P. 467-506.
21. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров. М.: ДМК Пресс, 2011.
22. А. В. Боресков и др. Предисл.: В. А. Садовничий. Параллельные вычисления на GPU. Архитектура и программная модель CUDA: Учеб. пособие. М.: МГУ, 2012.
23. Боресков А. В., Харламов А. А. Основы работы с технологией CUDA. М.: ДМК Пресс, 2012.
24. Сайт Лаборатории параллельных информационных технологий НИВЦ МГУ http://parallel.ru
25. Введение в OpenMP: <https://software.intel.com/ru-ru/blogs/2011/11/21/openmp-c?language=ru>
26. Учебное пособие по OpenMP <http://www.llnl.gov/computing/tutorials/openMP> (на англ. языке)
27. Евсеев И. MPI для начинающих: <http://www2.sscc.ru/Links/Litera/il/default.htm>
28. Официальный сайт NVIDIA CUDA <http://www.nvidia.ru/object/cuda_home_new_ru.html>
29. Официальная страница проекта LAMMPS: <http://lammps.sandia.gov> (на англ. языке)

## Справочники, словари, энциклопедии

1. Официальная документация по MPI: <http://www.mcs.anl.gov/mpi> (на англ. языке)
2. Официальная документация по OpenMP: http://www.openmp.org (на англ. языке)

## Программные средства

Для удаленного доступа к учебному вычислительному кластеру из компьютерного класса используются программы PuTTY (http://www.chiark.greenend.org.uk/~sgtatham/putty/) и WinSCP (http://winscp.net/). Также в процессе выполнения практических заданий студенты используют пакет программ молекулярно-динамического моделирования LAMMPS (http://lammps.sandia.gov/). Указанные программы являются свободно распространяемыми и не требуют приобретения лицензии.

## Дистанционная поддержка дисциплины

Предполагается дистанционное использование учебного вычислительного кластера для самостоятельно работы студентов во внеаудиторное время.

# Материально-техническое обеспечение дисциплины

Для выполнения практических заданий используется компьютерный класс с персональными компьютерами, на которых установлены операционные системы Windows или Linux, программы PuTTY и WinSCP для удаленного доступа к учебному вычислительному кластеру по сети Интернет. Также в учебном процессе частично задействованы два вычислительных кластера, один из которых содержит 26 процессорных ядер, другой (TEdge-48) – 192 процессорных ядра. На кластерах установлены ОС Linux, компиляторы с языков C/C++ и Fortran с поддержкой OpenMP, библиотека MPI, средства отладки, система очередей PBS, прикладной пакет атомистического моделирования LAMMPS.