



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

федеральное государственное бюджетное образовательное учреждение
высшего образования
**«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ИГУ»)**

Институт математики и информационных технологий
Кафедра алгебраических и информационных систем



Рабочая программа дисциплины (модуля)

Б1.О.11 Высокопроизводительные вычисления

Направление подготовки информационные технологии	02.04.02	Фундаментальная информатика и
Направленность (профиль) подготовки машинное обучение		Анализ данных научных исследований и
Квалификация выпускника	магистр	
Форма обучения	очная	

Иркутск 2025 г.

1. Цели и задачи дисциплины

Цель

Формирование компетенций в области применения современных высокопроизводительных вычислительных технологий в научных и прикладных исследованиях.

Задачи:

- Изучение архитектуры видеокарт Nvidia и их роли в научных расчетах;
- Формирование представлений о расширениях языков C/C++ для программирования видеокарт в технологии CUDA.
- Формирование умения разрабатывать SPMP-программы в технологии CUDA.
- Формирование представлений о работе параллельной программы, умений проектировать параллельные алгоритмы.
- Формирование умений применять возможности CUDA-библиотек машинного обучения (TensorRT, cuDNN, DeepStream SDK) в прикладных задачах.
- Изучение архитектуры и классификации многопроцессорных вычислительных систем;
- Формирование представлений о функциональности библиотеки mpi.h и её применении в программировании многопроцессорных систем.
- Формирование умений и навыков разработки программ в технологии MPI для кластеров и MPP систем
- Ознакомление с современными методами параллельных вычислений применительно к математическим и прикладным задачам.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

2.1. Учебная дисциплина (модуль) относится к обязательной части программы и изучается на первом курсе.

2.2. Для изучения данной учебной дисциплины (модуля) необходимы знания, умения и навыки, сформированные следующими дисциплинами: Б1.В.04 Машинное обучение, Б1.О.08 Математика для анализа данных.

2.3. Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной: Б1.В.06 Анализ и управление большими данными, Б2.О.03(П) Технологическая (проектно-технологическая), Б2.О.04(Пд) Преддипломная, Б3.01 Подготовка к сдаче и сдача государственного экзамена.

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование компетенций (элементов следующих компетенций) в соответствии с ФГОС ВО по соответствующему направлению подготовки.

Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение (в	ИДК опк2.1 Способен применять современное программное обеспечение (в том числе отечественного	Знает численные методы, алгоритмы применения которых поддаются распараллеливанию, сущность информационного графа алгоритма, основные функции CUDA и MPI, функциональность и ограничения библиотек CUDA для нейронных сетей и глубокого обучения, функциональность и ограничения

том числе отечественного производства) для решения задач профессиональной деятельности	производства) для решения задач профессиональной деятельности	инструментов анализа эффективности параллельных программ. Умеет интерпретировать математическую модель, выбирать численные методы решения, составлять информационный граф алгоритма в ярусно-параллельной форме, формулировать сопровождающее описание, реализовывать алгоритм в технологии CUDA и/или MPI с использованием подходящей библиотеки или без неё, анализировать эффективность своей параллельной программы. Владеет приемами проектирования параллельного алгоритма и разработки параллельных программ, методами оптимизации параллельного алгоритма, методами и инструментами анализа эффективности параллельных программ, разработкой параллельных алгоритмов для научных и прикладных задач, подходами к разработке параллельных алгоритмов для машинного обучения
	ИДК ОПК2.2 Способен применять суперкомпьютерные методы для решения задач профессиональной деятельности	

4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Объем дисциплины составляет 5 зачетных единиц, 180 часов, в том числе 26 часов на контроль, практическая подготовка 180 часов.
Форма промежуточной аттестации: 2 семестр - экзамен.

4.1. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, СТРУКТУРИРОВАННОЕ ПО ТЕМАМ, С УКАЗАНИЕМ ВИДОВ УЧЕБНЫХ ЗАНЯТИЙ И ОТВЕДЕННОГО НА НИХ КОЛИЧЕСТВА АКАДЕМИЧЕСКИХ ЧАСОВ

№ п/п	Раздел дисциплины/темы	Се мес тр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоя тельная работа + контроль	
			Лекции	Семинарск ие (практичес кие занятия)	Контроль обучения		
1.	Организация и системное программное обеспечение МВС	2	2	8	2	10	
1.1	Введение. Назначение и области применения высокопроизводительной вычислительной техники. Современная классификация суперкомпьютеров. TOP-500. Обоснование перспективности наращивания мощности ВВС. Закон Гроша. Закон Мура.	2	1			2	Самостоятельная работа
1.2	Архитектура высокопроизводительных ЭВМ. Определение. Классификация Флинна. Иерархическая организация памяти компьютера. Элементы архитектуры параллельных компьютеров. Связь между элементами параллельных вычислительных систем. Топология ВС. Масштабируемость. Статические топологии. Динамические топологии.	2	1			2	
1.3	Тема 3. Ресурсы вычислительного комплекса. Процессы. Синхронизация процессов. Распределение времени процессора. Управление памятью. Виртуальная память. Управление внешней памятью. Файловая система.	2		8		4	Практическая работа 1, промежуточное тестирование 1
2	Технология CUDA программирования графических процессоров.	2	4	8	2	36	
2.1	Архитектура графических процессоров nvidia	2	1			6	Подготовка доклада
2.2	Расширение языка C для CUDA: типы данных, директивы ядра, операции работы с памятью, операции обмена данными между CPU и GPU, сетка, блоки, нити, вызов вычислительного ядра.	2	1	4		10	Практическая работа 2, промежуточное тестирование 2

№ п/п	Раздел дисциплины/темы	Се мес тр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоя тельная работа + контроль	
			Лекции	Семинарск ие (практичес кие занятия)	Контроль обучения		
2.3	Виды памяти. Оптимизация работы с глобальной памятью.	2	1	2		8	Практическая работа 3, промежуточное тестирование 2
2.4	Библиотеки CUDA для работы с нейронными сетями и машинным обучением (TensorRT, Deep Stream)	2	1	2		12	Практическая работа 4, промежуточное тестирование 2
3.	Технология параллельного программирования MPI	2	6	8	3	40	
3.1	Введение в параллельное программирование с использованием стандарта MPI. Общая характеристика библиотеки MPI. Базовые понятия MPI: процесс, группа процессов, коммунитор. SPMD-модель передачи сообщений. Компиляция и запуск MPI-приложений. Структура MPI-программы.	2	1	0		2	Изучение мануала работы с кластером
3.2	Типы данных MPI. Синтаксис базовых функций MPI: инициализация и завершение, определение числа активированных процессоров и номера процесса в группе, передача сообщений между отдельными процессорами.	2	1	2		10	Практическая работа 5, промежуточное тестирование 3
3.3.	Двухточечный обмен сообщениями. Коллективный обмен данными в MPI	2	2	2		10	Практическая работа 6 промежуточное тестирование 3
3.4.	Управление областью взаимодействия и группой процессов. Производные типы данных.	2	1	2		6	Практическая работа 7, промежуточное тестирование 3
3.5	Параллельные математические библиотеки	2	1	2		12	Практическая работа 8,

№ п/п	Раздел дисциплины/темы	Се мес тр	Виды учебной работы, включая самостоятельную работу обучающихся и трудоемкость (в часах)				Формы текущего контроля успеваемости
			Контактная работа преподавателя с обучающимися			Самостоя тельная работа + контроль	
			Лекции	Семинарск ие (практичес кие занятия)	Контроль обучения		
							промежуточное тестирование 3
4	Анализ эффективности параллельных программ.	2	4	8	3	36	
4.1	Разработка эффективных параллельных программ: латентность, причины, сложности повышения эффективности	2	1			12	Самостоятельная работа (анализ программы)
4.2	Инструменты анализа эффективности: AIMS	2	2	4		12	Практическая работа 9
4.3	Инструменты анализа эффективности: Jumpshot	2	1	4		12	Практическая работа 10
Итого часов			16	32	10	122	

4.2. План внеаудиторной самостоятельной работы обучающихся по дисциплине

Семес тр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно- методическое обеспечение самостоятель ной работы
		Вид самостоятель ной работы	Сроки выполне ния	Затраты времени (час.)		
2	Организация и системное программное обеспечение МВС			10		

Семес тр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно- методическое обеспечение самостоятель ной работы
		Вид самостоятель ной работы	Сроки выполне ния	Затраты времени (час.)		
2	Введение. Назначение и области применения высокопроизводительной вычислительной техники. Современная классификация суперкомпьютеров. TOP-500. Обоснование перспективности наращивания мощности BBC. Закон Гроша. Закон Мура.	Для овладения знаниями: чтение, дополнительной литературы, конспекта лекций, поиск в сети Интернет; Для закрепления и систематизации знаний: работа с конспектом лекции; выполнение проектного задания, выполнение заданий для практических работ; Для формирования умений: решение ситуационных задач, сформулированных в практических работах; выполнение проектного задания	1 неделя	2	Задание для самостоятельной работы 1, 2	Материалы курса в ЭОС bki.forlabs.ru
2	Архитектура высокопроизводительных ЭВМ. Определение. Классификация Флинна. Иерархическая организация памяти компьютера. Элементы архитектуры параллельных компьютеров. Связь между элементами параллельных вычислительных систем. Топология ВС. Масштабируемость. Статические топологии. Динамические топологии.		1 неделя	2	Задание для самостоятельной работы 3	
2	Тема 3. Ресурсы вычислительного комплекса. Процессы. Синхронизация процессов. Распределение времени процессора. Управление памятью. Виртуальная память. Управление внешней памятью. Файловая система.		2 недели	4	Практическая работа 1	
2	Технология CUDA программирования графических процессоров.			36		
2	Архитектура графических процессоров nvidia		1 неделя	6	Задание для самостоятельной работы 4	Материалы курса в ЭОС bki.forlabs.ru
2	Расширение языка C для CUDA: типы данных, директивы ядра, операции работы с памятью, операции обмена данными между CPU и GPU, сетка, блоки, нити, вызов вычислительного ядра.		2 недели	8	Практическая работа 2	
2	Виды памяти. Оптимизация работы с глобальной памятью.		2 недели	10	Практическая работа 3	

Семес тр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно- методическое обеспечение самостоятельн ой работы
		Вид самостоятель ной работы	Сроки выполне ния	Затраты времени (час.)		
2	Библиотеки CUDA для работы с нейронными сетями и машинным обучением (TensorRT, Deep Stream)	проектного задания, выполнение заданий для практических работ; Для формирования умений: решение задач, сформулированных в практических работах; выполнение проектного задания	2 недели	12	Практическая работа 4 Задание для самостоятельной работы 5	Материалы курса в ЭОС bki.forlabs.ru
2	Технология параллельного программирования MPI	Для овладения		40		
2	Введение в параллельное программирование с использованием стандарта MPI. Общая характеристика библиотеки MPI. Базовые понятия MPI: процесс, группа процессов, коммуникатор. SPMD-модель передачи сообщений. Компиляция и запуск MPI-приложений. Структура MPI-программы.	Для овладения знаниями: чтение, дополнительной литературы, конспекта лекций, поиск в сети Интернет; Для закрепления и систематизации знаний: работа с конспектом лекции, выполнение заданий для практических работ; выполнение проектного задания Для формирования умений: решение задач, сформулированных в практических работах;	1 неделя	10	Задание для самостоятельной работы 6 Практическая работа 5	
2	Типы данных MPI. Синтаксис базовых функций MPI: инициализация и завершение, определение числа активированных процессоров и номера процесса в группе, передача сообщений между отдельными процессорами.		2 неделя	10	Практическая работа 6	
2	Двухточечный обмен сообщениями. Коллективный обмен данными в MPI		2 недели	6	Практическая работа 7	
2	Управление областью взаимодействия и группой процессов. Производные типы данных.					
2	Параллельные математические библиотеки		2 недели	12	Практическая работа 8 Задание для самостоятельной работы 7	
2	Анализ эффективности параллельных программ.			36		

Семес тр	Название раздела, темы	Самостоятельная работа обучающихся			Оценочное средство	Учебно- методическое обеспечение самостоятельн ой работы
		Вид самостоятель ной работы	Сроки выполне ния	Затраты времени (час.)		
2	Разработка эффективных параллельных программ: латентность, причины, сложности повышения эффективности	Для овладения знаниями: чтение, дополнительной литературы, конспекта лекций, поиск в сети Интернет, аудио и видеоматериалы рекомендованных МООК; Для закрепления и систематизации знаний: работа с конспектом лекции; выполнение заданий для практических работ; Для формирования умений: решение задач, сформулированных в практических работах;	1 неделя	12	Задание для самостоятельной работы 8	Материалы курса в ЭОС bki.forlabs.ru
2	Инструменты анализа эффективности: AIMS		2 недели	12	Практическая работа 9	
2	Инструменты анализа эффективности: Jumpshot		2 недели	12	Практическая работа 10	
Общая трудоемкость самостоятельной работы по дисциплине (час)				122		
Из них объем самостоятельной работы с использованием электронного обучения и дистанционных образовательных технологий (час)				106		

4.3. СОДЕРЖАНИЕ УЧЕБНОГО МАТЕРИАЛА

Содержание учебного материала прописано в таблице 4.1.

4.3.1. Перечень семинарских, практических занятий и лабораторных работ

№ п/п	№ раздела и темы	Наименование семинаров, практических и лабораторных работ	Трудоемкость (час.)		Оценочные средства	Формируемые компетенции (индикаторы)*
			Всего часов	Из них практическая подготовка		
1	2	3	4	5	6	7
	1.	Организация и системное программное обеспечение МВС	2	8		
	1.3	Ресурсы вычислительного комплекса. Процессы. Синхронизация процессов. Распределение времени процессора. Управление памятью. Виртуальная память. Управление внешней памятью. Файловая система.	2	8	Практическая работа 1	ОПК-2.1
	2	Технология CUDA программирования графических процессоров.	4	4		
	2.2	Расширение языка C для CUDA: типы данных, директивы ядра, операции работы с памятью, операции обмена данными между CPU и GPU, сетка, блоки, нити, вызов вычислительного ядра.	2	4	Практическая работа 2	ОПК-2.1, ОПК-2.2
	2.3	Виды памяти. Оптимизация работы с глобальной памятью.	1	2	Практическая работа 3	ОПК-2.1, ОПК-2.2
	2.4	Библиотеки CUDA для работы с нейронными сетями и машинным обучением (TensorRT, Deep Stream)	1	2	Практическая работа 4	ОПК-2.2, ОПК-2.3, ОПК-2.4
	3.	Технология параллельного программирования MPI	6	8		
	3.2	Типы данных MPI. Синтаксис базовых функций MPI: инициализация и завершение, определение числа активированных процессоров и номера процесса в группе, передача сообщений между отдельными процессорами.	2	2	Практическая работа 5	ОПК-2.1

	3.3.	Двухточечный обмен сообщениями. Коллективный обмен данными в MPI	2	2	Практическая работа 6	ОПК-2.1, ОПК-2.2
	3.4.	Управление областью взаимодействия и группой процессов. Производные типы данных.	1	2	Практическая работа 7	ОПК-2.1, ОПК-2.2, ОПК-2.3
	3.5	Параллельные математические библиотеки	1	2	Практическая работа 8	ОПК-2.3, ОПК-2.4
	4	Анализ эффективности параллельных программ.	4	8		
	4.2	Инструменты анализа эффективности: AIMS	2	4	Практическая работа 9	ОПК-2.3, ОПК-2.4
	4.3	Инструменты анализа эффективности: Jumpshot	2	4	Практическая работа 10	ОПК-2.3, ОПК-2.4
		Всего	16	16		

4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение студентами в рамках самостоятельной работы (СР)

№ п/п	Тема	Задание	Формируемая компетенция	ИДК
1.	Классификации многопроцессорных вычислительных систем, их сравнение	Задание для самостоятельной работы 1	ОПК-2	ОПК-2.1
2.	Рейтинг Top-500 и Top-50	Задание для самостоятельной работы 2	ОПК-2	ОПК-2.1
3.	Топологии МВС	Задание для самостоятельной работы 3	ОПК-2	ОПК-2.1
4.	Архитектура видеокарт NVIDIA разных поколений	Задание для самостоятельной работы 4	ОПК-2	ОПК-2.1
5.	Библиотека TensorRT	Задание для самостоятельной работы 5	ОПК-2	ОПК-2.2, ОПК-2.3, ОПК-2.4
6.	Библиотека Deep Stream	Задание для самостоятельной работы 5	ОПК-2	ОПК-2.2, ОПК-2.3, ОПК-2.4
7.	Основы управления параллельной программой на кластере	Задание для самостоятельной работы 6	ОПК-2	ОПК-2.1
8.	Параллельные математические библиотеки	Задание для самостоятельной работы 7	ОПК-2	ОПК-2.2, ОПК-2.3, ОПК-2.4
9.	Функциональность AIMS	Задание для самостоятельной работы 8	ОПК-2	ОПК-2.2, ОПК-2.3, ОПК-2.4
10.	Функциональность JumpShot	Задание для самостоятельной работы 8	ОПК-2	ОПК-2.2, ОПК-2.3, ОПК-2.4

4.4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

Методические указания по организации самостоятельной работы расположены в ИОС Educa. Самостоятельная работа студентов всех форм и видов обучения является одним из обязательных видов образовательной деятельности, обеспечивающей реализацию требований Федеральных государственных стандартов высшего профессионального образования. Согласно требованиям нормативных документов самостоятельная работа студентов является обязательным компонентом образовательного процесса, так как она обеспечивает закрепление получаемых на лекционных занятиях знаний путем приобретения навыков осмысления и расширения их содержания, навыков решения актуальных проблем формирования общекультурных и профессиональных компетенций, научно-исследовательской деятельности, подготовки к семинарам, лабораторным работам, сдаче зачетов и экзаменов. Самостоятельная работа студентов представляет собой совокупность аудиторных и внеаудиторных занятий и работ. Самостоятельная работа в рамках образовательного процесса в вузе решает следующие задачи:

- закрепление и расширение знаний, умений, полученных студентами во время аудиторных и внеаудиторных занятий, превращение их в стереотипы умственной и физической деятельности;
- приобретение дополнительных знаний и навыков по дисциплинам учебного плана;
- формирование и развитие знаний и навыков, связанных с научно-исследовательской деятельностью;
- развитие ориентации и установки на качественное освоение образовательной программы;
- развитие навыков самоорганизации;
- формирование самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации;
- выработка навыков эффективной самостоятельной профессиональной теоретической, практической и учебно-исследовательской деятельности.

Подготовка к лекции. Качество освоения содержания конкретной дисциплины прямо зависит от того, насколько студент сам, без внешнего принуждения формирует у себя установку на получение на лекциях новых знаний, дополняющих уже имеющиеся по данной дисциплине. Время на подготовку студентов к двухчасовой лекции по нормативам составляет не менее 0,2 часа.

Подготовка к практическому занятию. Подготовка к практическому занятию включает следующие элементы самостоятельной деятельности: четкое представление цели и задач его проведения; выделение навыков умственной, аналитической, научной деятельности, которые станут результатом предстоящей работы. Выработка навыков осуществляется с помощью получения новой информации об изучаемых процессах и с помощью знания о том, в какой степени в данное время студент владеет методами исследовательской деятельности, которыми он станет пользоваться на практическом занятии. Подготовка к практическому занятию нередко требует подбора материала, данных и специальных источников, с которыми предстоит учебная работа. Студенты должны дома подготовить к занятию 3–4 примера формулировки темы исследования, представленного в монографиях, научных статьях, отчетах. Затем они самостоятельно осуществляют поиск соответствующих источников, определяют актуальность конкретного исследования процессов и явлений, выделяют основные способы доказательства авторами научных работ ценности того, чем они занимаются. В ходе самого практического занятия студенты сначала представляют найденные ими варианты формулировки актуальности исследования, обсуждают их и обосновывают свое мнение о наилучшем варианте. Время на подготовку к практическому занятию по нормативам составляет не менее 0,2 часа.

Подготовка к экзамену. Самостоятельная подготовка к экзамену схожа с подготовкой к зачету, особенно если он дифференцированный. Но объем учебного материала, который нужно восстановить в памяти к экзамену, вновь осмыслить и понять, значительно больше, поэтому требуется больше времени и умственных усилий. Важно сформировать целостное представление о содержании ответа на каждый вопрос, что предполагает знание разных научных трактовок сущности того или иного явления, процесса, умение раскрывать факторы, определяющие их противоречивость, знание имен ученых, изучавших обсуждаемую проблему. Необходимо также привести информацию о материалах эмпирических исследований, что указывает на всестороннюю подготовку студента к экзамену. Время на подготовку к экзамену по нормативам составляет 36 часов.

Формы внеаудиторной самостоятельной работы

Составление глоссария Цель самостоятельной работы: повысить уровень информационный культуры; приобрести новые знания; отработать необходимые навыки в предметной области учебного курса. Глоссарий — словарь специализированных терминов и их определений. Статья глоссария — определение термина. Содержание задания: сбор и систематизация понятий или терминов, объединенных общей специфической тематикой, по одному либо нескольким источникам. Выполнение задания: 1) внимательно прочесть работу; 2) определить наиболее часто встречающиеся термины; 3) составить список терминов, объединенных общей тематикой; 4) расположить термины в алфавитном порядке; 5) составить статьи глоссария: — дать точную формулировку термина в именительном падеже; — объемно раскрыть смысл данного термина Планируемые результаты самостоятельной работы: способность студентов решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности.

Разработка проекта (индивидуального, группового) Цель самостоятельной работы: развитие способности прогнозировать, проектировать, моделировать. Проект — «ограниченное во времени целенаправленное изменение отдельной системы с установленными требованиями к качеству результатов, возможными рамками расхода средств и ресурсов и специфической организацией». Выполнение задания:

- 1) диагностика ситуации (проблематизация, целеполагание, конкретизация цели, форматирование проекта);
- 2) проектирование (уточнение цели, функций, задач и плана работы; теоретическое моделирование методов и средств решения задач; детальная проработка этапов решения конкретных задач; пошаговое выполнение запланированных проектных действий; систематизация и обобщение полученных результатов, конструирование предполагаемого результата, пошаговое выполнение проектных действий);
- 3) рефлексия (выяснение соответствия полученного результата замыслу; определение качества полученного продукта; перспективы его развития и использования).

Предполагаемые результаты самостоятельной работы: готовность студентов использовать знание современных проблем науки и образования при решении образовательных и профессиональных задач; готовность использовать индивидуальные креативные способности для оригинального решения исследовательских задач; — способность прогнозировать, проектировать, моделировать.

Информационный поиск Цель самостоятельной работы: развитие способности к проектированию и преобразованию учебных действий на основе различных видов информационного поиска. Информационный поиск — поиск неструктурированной документальной информации. Список современных задач информационного поиска: решение вопросов моделирования; классификация документов; фильтрация, классификация документов; проектирование архитектур поисковых систем и пользовательских интерфейсов; извлечение информации

(аннотирование и реферирование документов); выбор информационно-поискового языка запроса в поисковых системах. Содержание задания по видам поиска: поиск библиографический — поиск необходимых сведений об источнике и установление его наличия в системе других источников. Ведется путем разыскания библиографической информации и библиографических пособий (информационных изданий); поиск самих информационных источников (документов и изданий), в которых есть или может содержаться нужная информация; — поиск фактических сведений, содержащихся в литературе, книге (например, об исторических фактах и событиях, о биографических данных из жизни и деятельности писателя, ученого и т. п.). Выполнение задания:

- 1) определение области знаний;
- 2) выбор типа и источников данных;
- 3) сбор материалов, необходимых для наполнения информационной модели;
- 4) отбор наиболее полезной информации;
- 5) выбор метода обработки информации (классификация, кластеризация, регрессионный анализ и т.д.);
- 6) выбор алгоритма поиска закономерностей;
- 7) поиск закономерностей, формальных правил и структурных связей в собранной информации;
- 8) творческая интерпретация полученных результатов.

Планируемые результаты самостоятельной работы: — способность студентов решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; готовность использовать знание современных проблем науки и образования при решении образовательных и профессиональных задач.

Разработка мультимедийной презентации

Цели самостоятельной работы:

- освоение (закрепление, обобщение, систематизация) учебного материала;
- обеспечение контроля качества знаний;
- формирование специальных компетенций, обеспечивающих возможность работы с информационными технологиями;
- становление общекультурных компетенций.

Выполнение задания:

1. Этап проектирования: — определение целей использования презентации; — сбор необходимого материала (тексты, рисунки, схемы и др.); — формирование структуры и логики подачи материала; — создание папки, в которую помещен собранный материал.

2. Этап конструирования:

- выбор программы;
- определение дизайна слайдов;
- наполнение слайдов собранной текстовой и наглядной информацией;
- включение эффектов анимации и музыкального сопровождения (при необходимости);
- установка режима показа слайдов (титовый слайд, включающий наименование кафедры, где выполнена работа, название презентации, город и год; содержательный — список слайдов презентации, сгруппированных по темам сообщения; заключительный слайд

содержит выводы, пожелания, список литературы и пр.).

3. Этап моделирования — проверка и коррекция подготовленного материала, определение продолжительности его демонстрации.

Планируемые результаты самостоятельной работы:

— повышение информационной культуры студентов и обеспечение их готовности к интеграции в современное информационное пространство;

— способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности;

— способность к критическому восприятию, обобщению, анализу профессиональной информации, постановке цели и выбору путей ее достижения; — способность применять современные методики и технологии организации и реализации образовательного процесса на различных образовательных ступенях в различных образовательных учреждениях;

— готовность использовать индивидуальные креативные способности для оригинального решения исследовательских задач.

В ФБГОУ ВО «ИГУ» организация самостоятельной работы студентов регламентируется Положением о самостоятельной работе студентов, принятым Ученым советом ИГУ 22 июня 2012 г.

4.5. ПРИМЕРНАЯ ТЕМАТИКА КУРСОВЫХ РАБОТ (ПРОЕКТОВ)

Не предусмотрено.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) перечень литературы

1. Гергель, В. П. Теория и практика параллельных вычислений : учебное пособие / В. П. Гергель. — 2-е изд. — Москва : ИНТУИТ, 2016. — 500 с. — ISBN 978-5-94774-645-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/100527> . — Режим доступа: для авториз. пользователей.

2. Малявко, А. А. Параллельное программирование на основе технологий OpenMP, MPI, CUDA : учебное пособие / А. А. Малявко. — Новосибирск : НГТУ, 2015. — 116 с. — ISBN 978-5-7782-2614-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118245> . — Режим доступа: для авториз. пользователей.

3. Филиппов, Ф. В. Многоядерные процессоры и параллельное программирование : учебное пособие / Ф. В. Филиппов, А. Н. Губин. — Санкт-Петербург : СПбГУТ им. М.А. Бонч-Бруевича, 2013. — 99 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/180060> . — Режим доступа: для авториз. пользователей.

б) периодические издания

1. CODATA Data Science Journal <https://datascience.codata.org/> CODATA Data Science Journal - это рецензируемый электронный журнал с открытым доступом, публикующий статьи по управлению, распространению, использованию и повторному использованию исследовательских данных и баз данных во всех

в) список авторских методических разработок:

Справочные материалы и задания в ЭОС bki.forlabs.ru

г) базы данных, информационно-справочные и поисковые системы

1. Единое окно доступа к образовательным ресурсам. Полнотекстовая электронная библиотека учебных и учебно-методических материалов (федеральный ресурс). <http://www.window.edu.ru>.

2. Образовательный математический сайт. [http:// www.exponenta.ru](http://www.exponenta.ru).

3. Московский центр непрерывного математического образования, МЦНМО. Материалы (полные тексты) свободно распространяемых книг по математике. <http://www.mcsme.ru/free-books>.

4. База знаний и набор вычислительных алгоритмов. <http://www.wolframalpha.com>.

5. Электронно-библиотечная система издательства «ЮРАЙТ» <https://www.biblio-online.ru/>

6. Электронно-библиотечная система издательства «Лань» <https://e.lanbook.com/>

7. ИОС ИГУ EDuCa

8. Онлайн-курсы от ведущих вузов и компаний страны <https://welcome.stepik.org/ru>

9. Образовательный онлайн-проект <https://www.coursera.org/>

10. KD Nuggets — крупнейший ресурс о науке о данных, где собраны знания для всех уровней: от начинающих специалистов до профессиональных инженеров.

11. Indigo — внедрение data science-решений на примере конкретных кейсов, рассказанных компаниями-разработчиком.

12. <http://www.parallel.ru> — Информационно-аналитические материалы по параллельным вычислениям.

13. <http://www.software.unn.ac.ru/ccam> — Информационные материалы Центра компьютерного моделирования Нижегородского университета.

14. <http://www.ieeetfcc.org> — Информационные материалы рабочей группы IEEE по кластерным вычислениям.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. УЧЕБНО-ЛАБОРАТОРНОЕ ОБОРУДОВАНИЕ:

Для проведения лекционных занятий необходима аудитория с презентационным оборудованием, для проведения практических занятий необходима аудитория на 15-30 рабочих мест (в зависимости от численности учебной группы), оборудованная доской, презентационной техникой.

6.2. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ:

№	Наименование Программного продукта	Кол-во	Обоснование для пользования ПО	Дата выдачи лицензии	Срок действия права пользования
1.	Java 8	Условия правообладателя	Условия использования по ссылке: https://www.oracle.com/legal/terms.html	Условия правообладателя	бессрочно
2.	Joomla 3.6	Условия правообладателя	Условия использования по ссылке: https://docs.joomla.org/JEDL	Условия правообладателя	бессрочно
3.	Microsoft Office Professional Plus 2007 Russian Academic OPEN No Level	25	Номер Лицензии Microsoft 46211164 Гос.контракт № 03-162-09 от 01.12.2009	01.12.2009	бессрочно
4.	Microsoft Office Professional Plus 2007 Russian Academic OPEN No Level	10	Номер Лицензии Microsoft 42095516	27.04.2007	бессрочно
5.	UbuntuLinux 16.04.1	Условия правообладателя	Условия использования по ссылке: https://www.ubuntu.com/legal/terms-and-policies/terms	Условия правообладателя	бессрочно

6.3. ТЕХНИЧЕСКИЕ И ЭЛЕКТРОННЫЕ СРЕДСТВА:

презентационное оборудование, персональный компьютер с возможностью демонстрации презентаций в формате pdf, персональные компьютеры с установленным ПО, описанным в п. 6.2.

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

При реализации программы данной дисциплины используются различные образовательные технологии.

1. Проблемное обучение. Создание в учебной деятельности проблемных ситуаций и организация активной самостоятельной деятельности учащихся по их разрешению, в результате чего происходит творческое овладение знаниями, умениями, навыками, развиваются мыслительные способности
2. Разноуровневое обучение. У преподавателя появляется возможность помогать слабому, уделять внимание сильному, реализуется желание сильных учащихся быстрее и глубже продвигаться в образовании. Сильные учащиеся утверждаются в своих способностях, слабые получают возможность испытывать учебный успех, повышается уровень мотивации учения.
3. Проектные методы обучения. Работа по данной методике дает возможность развивать индивидуальные творческие способности учащихся, более осознанно подходить к профессиональному и социальному самоопределению
4. Исследовательские методы в обучении. Дает возможность учащимся

самостоятельно пополнять свои знания, глубоко вникать в изучаемую проблему и предполагать пути ее решения, что важно при формировании мировоззрения. Это важно для определения индивидуальной траектории развития каждого обучающегося

5. Лекционно-семинарско-зачетная система. Данная система дает возможность сконцентрировать материал в блоки и преподносить его как единое целое, а контроль проводить по предварительной подготовке обучающихся

6. Информационно-коммуникационные технологии. Изменение и неограниченное обогащение содержания образования, использование интегрированных курсов, доступ в ИНТЕРНЕТ.

Наименование тем занятий с использованием активных форм обучения:

№	Тема занятия	Вид занятия	Форма / Методы интерактивного обучения	Кол-во часов
1.	Построение двумерных сеток в программах на CUDA	Практическая работа	Проблемная ситуация, работа в малых группах	2
2.	TensorRT, Deep Stream	Практическая работа	Кейс-метод, работа в малых группах	2
3.	Математические параллельные библиотеки MPI	Практическая работа	Метод анализа конкретных ситуаций	2
4.	Анализ эффективности параллельных программ	Практическая работа	Проблемная ситуация, работа в малых группах	2
Итого часов				8

8. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

8.1. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ВХОДНОГО КОНТРОЛЯ

Не предусмотрены.

8.2. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ТЕКУЩЕГО КОНТРОЛЯ

Практические работы

Практическая работа 2. Основы CUDA

Часть 1. Взаимодействие с кластером

Выполнить компиляцию и запуск на кластере приведённой в презентации программы тестирования устройства и вывода на экран технических характеристик видеокарты. Результаты работы программы оформить в виде таблицы в файле формата *.docx или *.pdf.

Для входа на кластер можно использовать программу putty (для ОС Windows) или командную строку в ОС Linux. Протокол SSH.

IP: 194.176.114.13

login:

password:

Команда подключения в ОС Linux: `ssh <login>@<ip> <password>`

mc - вызов приложения Midnight Commander

Исходный код программы для видеокарты сохраняется с расширением *.cu

Для компиляции используется команда

`>> nvcc <имя_исходника> -o <имя_объектного_файла>`

Для запуска - ./имя_файла

Часть 2. Ядро CUDA, передача данных на видеокарту

Дополнить разобранный на занятии программу пересчета данных одномерного массива (см. example1.cu) следующим образом:

1. Замерить время работы параллельной программы:

а) с учетом передачи данных с хоста на девайс и обратно

- б) только время работы ядра CUDA
2. Найти способ замера времени в **миллисекундах** для последовательной программы.
 3. Написать аналогичную последовательную программу и замерить время её работы (только вычислительной части, время формирования исходных данных не учитывать)
 4. Сравнить результаты трёх замеров времени.
 5. Прикрепить коды параллельной и последовательной программ. В коде параллельной программы в комментариях указать результаты замеров. Комментарии с замерами расположить в начале файла.

Часть 3. Одномерные сетки

1. Напишите параллельную программу сложения двух массивов достаточного объема.
2. Замерьте время работы ядра CUDA вашей программы.
3. Видоизмените сетку и замерьте время еще раз.
4. Напишите аналогичную последовательную программу, замерьте время работы основного расчетного цикла (обратите внимание, что замерять время надо в **миллисекундах**, а стандартная библиотека `time.h` выполняет замеры в секундах, т.е. надо найти необходимые функции). Сравните время последовательной программы и наиболее быстрой параллельной.

Практическая работа 3. Виды памяти видеокарты и работа с ними

Часть 1. Глобальная память

1. Написать параллельную программу перемножения матриц
 2. Написать параллельную программу решения СЛАУ методом простых итераций (использование `coalescing` добавляет баллы)
 3. Замерить время работы ядра каждой программы
- Программы размещать в своей папке на кластере.

Часть 2. Разделяемая память

Реализовать рассмотренные на лекции алгоритмы перемножения матриц и редукции с использованием разделяемой памяти.

Замерить время работы каждого алгоритма и сравнить со временем работы такого же алгоритма, но без использования `shared memory`

Практическая работа 4. TensorRT

Из электронной среды скачать веса предобученной модели распознавания тайлов игры Маджонг, скачать сетки данных.

Используя веса и данные, написать программу распознавания тайлов с использованием библиотеки TensorRT

Практическая работа 5.

Параллельное программирование на основе MPI.

Инструкция к запуску и компиляции MPI-программы на кластере cluster-sfu.krasn.ru.

1. Зайдите на кластер под своим логином.
2. Загрузите Midnight Commander командой `mc`
3. Создайте папку для хранения программ клавишей F7, перейдите в созданную папку.
4. Загрузите в созданную папку исходный код программы с помощью WinSCP или создайте новый файл в Midnight Commander, нажав Shift+F4 и отредактировав его в текстовом редакторе `mc` (переход в текстовый редактор F4)
5. Наберите текст программы, выводящей сообщения (Hello, world!!) от процессов (пример из лекции 14). Для сохранения изменений в текстовом редакторе `mc` используйте F2.
6. Для компиляции программы:
 - a. Сверните панели `mc` (ctrl+o)
 - b. Выполните компиляцию файла программы командой: `mpicc -o <имя_объектного_файла> <имя_компилируемого_файла>`. Или `mpiCC -o <имя_объектного_файла> <имя_компилируемого_файла>`. Если

компиляция выполнена без ошибок, то в папке появится объектный файл (выделен зеленым цветом)

7. Запустите программу на выполнение командой: **mpirun -np <количество_процессов> -maxtime 1 <имя_объектного_файла>**

Задачи (для «опытных»).

1. Написать программу, которая передаёт некоторое число по кольцу: от процесса с рангом 0, на процесс 1, процесс 1 прибавляет к числу 1 и передаёт на процесс 2 и т.д. Полученное в результате число выводится на консоль процессом с рангом 0. Замерить время выполнения программы
2. Составить параллельную программу подсчёта суммы нечётных элементов вектора. Замерить время выполнения программы. Использовать функции широковещательной рассылки и редукции. *Использовать функцию распределения данных MPI_Scatter(...) или MPI_Scatterv(...)

Для каждой задачи:

- a. Запустить программу с одной и той же размерностью вектора несколько раз на 2-х, 4-х, 8-ми, 16-ти и 32-х процессорах с фиксацией времени выполнения программы при каждом запуске.
- b. Построить в Excel график зависимости скорости выполнения программы от количества процессоров.
- c. Запустить программу несколько раз на 16 процессорах, изменяя размерность массива в несколько раз при каждом запуске. Зафиксировать время выполнения
- d. Построить в Excel график зависимости скорости выполнения программы от увеличения количества входных данных при одном и том же количестве процессоров

Задачи (для «начинающих»).

1. Написать параллельную программу «Hello, world!!»
2. Написать программу, которая передаёт некоторое число по кольцу: от процесса с рангом 0, на процесс 1, процесс 1 прибавляет к числу 1 и передаёт на процесс 2 и т.д. Полученное в результате число выводится на консоль процессом с рангом 0. Замерить время выполнения программы

Для каждой задачи (кроме «Hello, world»):

- a. Запустить программу с одной и той же размерностью вектора несколько раз на 2-х, 4-х, 8-ми, 16-ти и 32-х процессорах с фиксацией времени выполнения программы при каждом запуске.
- b. Построить в Excel график зависимости скорости выполнения программы от количества процессоров.
- c. Запустить программу несколько раз на 16 процессорах, изменяя размерность массива в несколько раз при каждом запуске. Зафиксировать время выполнения
- d. Построить в Excel график зависимости скорости выполнения программы от увеличения количества входных данных при одном и том же количестве процессоров

Практическая работа 6. Парные и коллективные операции передачи сообщений.

Часть 1. Парные операции передачи данных

Для «начинающих»

Задание 1.

2. Для программы передачи числа по кольцу (см. предыдущую работу). При написании программы:
 - a. Использовать стандартный режим передачи сообщений
 - b. Добавить расчёт времени работы программы, зафиксировать время работы программы на 20 процессорах.
3. Изменить режим передачи данных на буферизованный и зафиксировать время работы программы на 20 процессорах.

4. Изменить режим передачи данных на режим по готовности и зафиксировать время работы программы на 20 процессах
5. Изменить режим передачи данных на синхронный режим и зафиксировать время работы программы на 20 процессах
6. Определить наиболее быстрый и наиболее медленный режимы работы программы.

Задание 2.

1. Написать программу суммирования элементов вектора, в которой:
 - а. должна быть дописана функция заполнения исходного массива (случайными числами или согласно какой-либо математической формуле);
 - б. должен быть добавлен расчет времени выполнения программы
2. Запустить программу с одной и той же размерностью вектора несколько раз на 2-х, 4-х, 8-ми, 16-ти и 32-х процессорах с фиксацией времени выполнения программы при каждом запуске.
3. Построить в Excel график зависимости скорости выполнения программы от количества процессоров.
4. Запустить программу несколько раз на 16 процессорах, изменяя размерность массива в несколько раз при каждом запуске. Зафиксировать время выполнения
5. Построить в Excel график зависимости скорости выполнения программы от увеличения количества входных данных при одном и том же количестве процессоров

Для «опытных»

Задача 1.

Написать программу перемножения двух матриц. *Использовать функцию распределения данных MPI_Scatter(...) или MPI_Scatterv(...)

Задача 2.

Написать программу перемножения двух матриц, при условии, что в оперативную память процессора ни одна из матриц полностью не помещается. Т.е. в один и тот же момент времени на процессоре есть несколько строк первой матрицы, несколько столбцов второй матрицы и несколько строк результирующей матрицы. Для решения задачи исходные данные должны передаваться от процесса к процессу по любому алгоритму так, чтобы в итоге все результирующие элементы оказались на своих местах на нужных процессорах. Для упрощения решения считать, что при определении количества строк (столбцов), хранимых на процессе ($k = N/p$), памяти на процессе достаточно для хранения данных и результатов.

Творческое задание. Для интересующихся.

Генетические алгоритмы.

1. Познакомиться с материалами по естественным моделям параллельных вычислений (лекция МГУ, код последовательной программы).
2. Проверить работу последовательного алгоритма
3. Реализовать параллельный генетический алгоритм непрерывной минимизации n -мерной непрерывной функции (найти точку глобального минимума). Функция Rosenbrock's valley:

$$f(x) = \sum_{i=1}^{n-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2].$$

где $n = 10^3$, $x_i \in [-100, 100]$, $size = 2^{12}$, $tmax = 10^4$, $dt = 10^2$, $fraction = 4$

Результаты:

- 1) среднее и лучшее значение функции качества от номера итерации t ;
- 2) время T работы программы от количества процессоров p .
4. Построить графики зависимости времени работы алгоритма от количества процессов параллельной программы
- 5.* Реализовать распределенную модель миграции

Часть 2. Коллективные операции передачи данных

Задание 1. Учебное

7. Реализовать программу умножения матрицы на вектор с разбиением матрицы на неравные части и рассылку частей на каждый процесс с помощью функции MPI_Scatterv(...). Провести серию вычислительных экспериментов и сравнить результаты с версией программы без коллективных операций:
- Запустить программу с одной и той же размерностью матрицы и вектора несколько раз на 2-х, 4-х, 8-ми, 16-ти и 32-х процессорах с фиксацией времени выполнения программы при каждом запуске.
 - Построить в Excel график зависимости скорости выполнения программы от количества процессоров.
 - Запустить программу несколько раз на 16 процессорах, изменяя размерность массивов в несколько раз при каждом запуске. Зафиксировать время выполнения
 - Построить в Excel график зависимости скорости выполнения программы от увеличения количества входных данных при одном и том же количестве процессоров

Задание 2. Творческое. Определить, сможет ли космический корабль пройти через астероидное поле.

Упрощения:

- Рассматривается задача на плоскости
- Частицы движутся по круговым орбитам
- Столкновения частиц и гравитационное взаимодействие в модели не учитываются
- Скорость каждой частицы считать постоянной

Условие задачи.

Есть «рой частиц», названных астероидами. Астероиды вращаются по круговым орбитам вокруг некоторой центральной области. Начало координат находится в центре области (рис. 1).

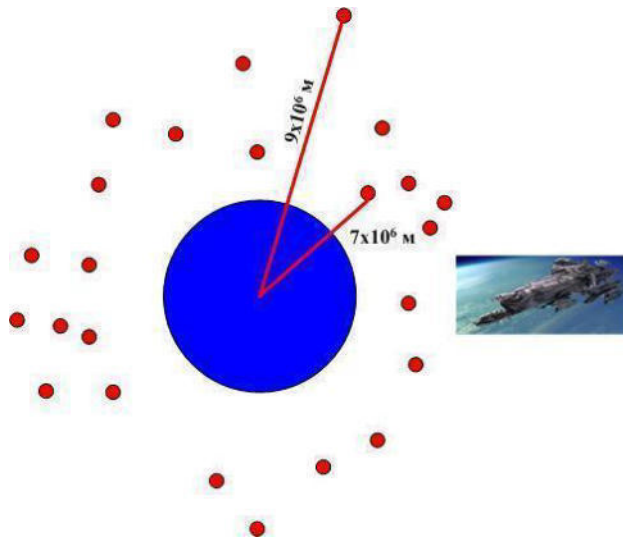


Рис. 1. Общая схема задачи

Каждая частица характеризуется (рис 2.):

- φ – угол между положительным направлением оси OX и отрезком, проведенным из начала координат к местоположению частицы;
- r – расстояние до частицы от начала координат;
- v м/с – скорость движения частицы вокруг их общего центра

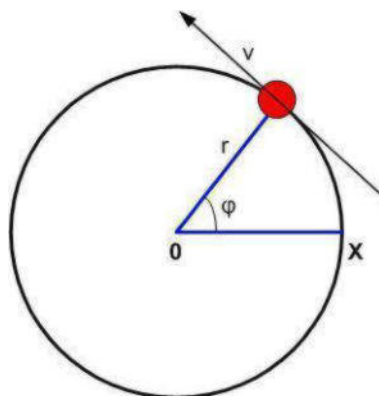


Рис.2. Характеристики частицы

Есть крупное (по сравнению с астероидами) космическое тело – корабль. Корабль характеризуется:

- 1) φ_s – угол между положительным направлением оси ОХ и отрезком, проведенным из начала координат к местоположению корабля;
- 2) r_s – расстояние до корабля от начала координат;
- 3) v_s м/с – скорость движения корабля.

Задача: определить, сможет ли корабль пройти сквозь рой частиц без повреждений.

Указания:

1. Количество частиц не менее 5×10^4 .
2. Решение задачи может быть построено от готового варианта прохождения корабля, т.е. можно сгенерировать конфигурацию частиц и корабля, которая решает задачу, а затем выполнить серию обратных преобразований и получить исходные условия. Обратную «прокрутку» выполнить не менее, чем за условный час.
3. Местоположение частиц сгенерировать, исходя из указания 2. Скорости частиц генерируются случайно в диапазоне $4000 \text{ м/с} \pm 5\%$.
4. Радиус орбиты астероидов должен быть в диапазоне от 7×10^6 м до 9×10^6 м

Практическая работа 7. Производные типы данных в MPI. Создание коммунитаторов.

Часть 1.

1. Процессы разбить на две группы. Одна содержит процессы с чётными рангами, а другая – с нечётными.
2. На базе каждой группы создать соответствующий коммунитатор
3. Организовать передачу по кольцу в каждом коммунитаторе

Часть 2.

Создать собственный тип данных для пересылки каждого третьего элемента массива.

2. Выполнить пересылку
3. Убедиться, что адресату пришли только нужные элементы

Практическая работа 8. Параллельные математические библиотеки.

Реализовать параллельный алгоритм быстрого преобразования Фурье для дискретизации аналогового звукового сигнала с использованием библиотеки FFT

Практическая работа 9. Инструменты анализа эффективности параллельной программы. AIMS.

Проанализировать с помощью AIMS эффективность параллельной программы решения СЛАУ методом Гаусса. Найти точки потери производительности. Предложить варианты устранения недостатков.

Практическая работа 10. Инструменты анализа эффективности параллельной программы. Jumpshot.

Проанализировать с помощью JumpShot эффективность параллельной программы решения СЛАУ методом Якоби. Найти точки потери производительности. Предложить варианты устранения недостатков.

Самостоятельные работы.

Задание для самостоятельной работы 1. Классификации МВС, топологии, рейтинги

Подготовить доклад на одну из тем:

1. Рейтинг Top-500 лучших высокопроизводительных систем мира
2. Рейтинг Green-500 лучших энергоэкономичных вычислительных систем мира
3. Рейтинг Top-50 лучших высокопроизводительных систем СНГ
4. Кластер Ломоносов-2. История, архитектура, возможности.
5. Серия суперкомпьютеров BlueGene
6. Физические задачи, решаемые на высокопроизводительных системах

Задание для самостоятельной работы 2. Эффективность параллельного алгоритма

1. Оцените ускорение и эффективность предложенного алгоритма вычисления скалярного произведения двух векторов
2. Пусть нужно ускорить работу программы в 10 раз. 1/10 программы можно ускорить не более чем в 5 раз. Во сколько раз нужно ускорить оставшиеся 9/10 программы, чтобы достичь цели

Задание для самостоятельной работы 3. Ярусно-параллельная форма графа параллельного алгоритма.

1. Предложите параллельный алгоритм поиска скалярного произведения двух векторов. Опишите каждый этап построения алгоритма.
2. Нарисуйте информационный граф для последовательного и для параллельного алгоритмов

Задание для самостоятельной работы 4. Архитектура видеокарт.

Подготовить доклад об архитектуре видеокарт поколений:

- Kepler
- Maxwell
- Pascal
- Turing
- Volta

В докладе обратить внимание на Compute Capability, описать возможности видеокарт, добавленные в рассматриваемой архитектуре.

Для доклада выбрать одну из перечисленных архитектур. Основной источник информации: сайт nvidia.com

Задание для самостоятельной работы 5. Библиотеки NVIDIA для работы с нейронными сетями и машинного обучения.

Изучить документацию по TensorRT SDK. Найти примеры применения библиотеки. Взять нейронную, обученную в курсе «Машинное обучение» и переработать с использованием TensorRT.

Задание для самостоятельной работы 6. Взаимодействие с кластером.

Изучить мануал пользователя для работы с кластером. Протестировать все команды на примере учебной задачи.

Задание для самостоятельной работы 7. Параллельные математические библиотеки.

Реализовать интерполирование данных многочленом Лагранжа с использованием библиотеки LAPACK. Данные для интерполирования в электронной учебной системе.

Задание для самостоятельной работы 8. Анализ эффективности параллельных программ.

Изучить документацию по AIMS и JumpShot. Проанализировать этими инструментами программы, написанные при выполнении практической работы 6. Выявить точки потери производительности. Предложить способы улучшения

8.3. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПРОМЕЖУТОЧНОГО КОНТРОЛЯ

№ п/п	Вид контроля	Контролируемые темы (разделы)	Компетенции, компоненты которых контролируются
	Промежуточное тестирование 1	Архитектура MVC	ОПК-2
	Промежуточное тестирование 2	Технология CUDA	
	Промежуточное тестирование 3	Технология MPI	

Оценка на экзамене выставляется как среднее арифметическое оценок, полученных в результате прохождения теста и оценки, полученной на практике.

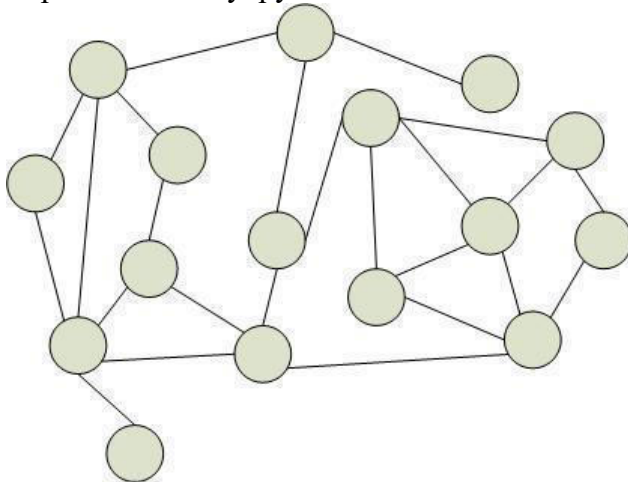
Демонстрационный вариант теста

- Отметьте, какие утверждения о массивно-параллельных компьютерах верны?
 - массивно-параллельные компьютеры не могут работать без хост-машины;
 - основными отличительными характеристиками массивно-параллельных компьютеров являются: используемые микропроцессоры и коммуникационная сеть (среда);
 - в массивно-параллельных компьютерах не может быть больше 1024 процессоров;
 - коммуникационная сеть массивно-параллельных компьютеров всегда организуется в виде трёхмерного тора.
- Отметьте верные утверждения о кластерных вычислительных системах:
 - кластерные системы строятся на базе серийных процессоров;
 - на каждом узле кластера исполняется свой экземпляр операционной системы;
 - максимально возможное число процессоров кластерной системы равно 128;
 - один кластер может строиться с использованием нескольких коммуникационных технологий.
- Разновидностью распараллеливания являются технологии и приёмы:
 - суперскалярности;
 - структурного программирования;
 - многопроцессорности;
 - объектно-ориентированного программирования.
- Отметьте верные утверждения:
 - MPI - это сокращение от Message Passing Interface.
 - Функция MPI_Comm_size определяет общее число запущенных параллельных процессов приложения.
 - Каждый параллельный процесс в MPI имеет номер.
 - MPI - это сокращение от Multiple Parallel Interface.
- Какая операция MPI не относится к коллективным операциям?
 - MPI_Bcast;
 - MPI_Send;
 - MPI_Reduce;
 - MPI_Gather.
- Отметьте верные утверждения:
 - использование функций MPI_Send и MPI_Recv может привести к тупиковой ситуации (deadlock);

- b) в коллективных операциях участвуют все процессы приложения;
 - c) функция, соответствующая коллективной операции, должна быть вызвана каждым процессом, быть может, со своим набором параметров;
 - d) возврат процесса из функции, реализующей коллективную операцию, всегда означает, что операция уже завершена;
7. Фрагмент программы:
- ```
MPI_Comm_size(comm, &size);
MPI_Send(buf, 15, MPI_INT, size+1, 8, comm);
MPI_Send(buf, 8, MPI_INT, size-1, 15, comm);
```
- a) может быть правильным;
  - b) заведомо содержит одну ошибку;
  - c) заведомо содержит две ошибки;
  - d) заведомо содержит три ошибки.
8. Отметьте верные утверждения о функции MPI\_Recv:
- a) перед вызовом функции MPI\_Recv надо обратиться к функции MPI\_Get\_count;
  - b) возврат из функции означает, что либо произошла ошибка, либо принятое сообщение расположено в первом параметре;
  - c) функцией MPI\_Recv нельзя принимать сообщение, посланное с помощью функции MPI\_Ssend;
  - d) нельзя использовать функцию MPI\_Recv, если неизвестны отправитель сообщения или тег сообщения.
9. Посылка сообщения с блокировкой (MPI\_Send) означает, что возврат из функции произойдёт тогда, когда:
- a) можно повторно использовать параметры данной функции;
  - b) сообщение покинет процесс;
  - c) сообщение принято адресатом;
  - d) адресат инициировал приём данного сообщения.
10. Функция MPI\_Send возвращает:
- a) число байт, заявленное в вызове для пересылки;
  - b) значение MPI\_SUCCESS или код ошибки;
  - c) номер процесса, которому адресована передача;
  - d) число реально переданных байт.
11. В декартовой топологии множество процессов представляется в виде:
- a) прямоугольной решётки;
  - b) графа произвольного вида;
  - c) полного графа;
  - d) звезды.
12. MPI поддерживает топологии вида:
- a) прямоугольная решётка произвольной размерности и граф произвольного вида;
  - b) только прямоугольная решётка произвольной размерности;
  - c) только граф произвольного вида;
  - d) тор произвольной размерности и граф произвольного вида.
13. Операция широковещательной рассылки данных это:
- a) операция рассылки значений ведущим процессом всем остальным процессам коммутатора; все процессы получают часть исходных данных;
  - b) операция рассылки значений ведущим процессом всем остальным процессам коммутатора, все процессы получают рассылаемые данные целиком;
  - c) операция рассылки различающихся значений ведущим процессом всем остальным процессам коммутатора;
  - d) операция рассылки значений ведущим процессом всем остальным процессам приложения, все процессы получают рассылаемые данные целиком.
14. Операцию редукции данных MPI\_Reduce можно описать:

- a) операцию передачи данных, при которой над собираемыми значениями осуществляется та или иная обработка, при этом результат обработки получают все процессы коммуникатора;
  - b) как операцию передачи данных, при которой над собираемыми значениями осуществляется обработка, при этом частичные значения результатов редуцирования получают все процессы параллельной программы;
  - c) как операцию передачи данных, при которой над собираемыми значениями осуществляется обработка в процессе передачи, при этом результат обработки получает только ведущий процесс;
  - d) операция передачи данных, при которой все процессы коммуникатора получают различающиеся значения.
15. В коллективных операциях передачи данных обязаны принимать участие:
- a) все процессы программы;
  - b) все процессы группы коммуникаторов;
  - c) все процессы одного коммуникатора;
  - d) некоторые процессы одного коммуникатора.
16. Режим передачи по готовности может быть использован только если:
- a) операция приёма сообщения уже инициирована;
  - b) операция приёма сообщения гарантированно будет запущена позднее момента начала передачи сообщения;
  - c) при размере сообщения, меньшем размера системного буфера.
17. В буферизованном режиме функция отправки сообщения завершается:
- a) сразу же после копирования сообщения в системный буфер;
  - b) при получении от процесса-получателя подтверждения о начале приёма отправленного сообщения;
  - c) при начале фактической передачи сообщения.
18. В синхронном режиме передачи завершение функции отправки сообщения происходит:
- a) при старте передачи данных процессом-отправителем по сети;
  - b) при завершении копирования сообщения в системный буфер;
  - c) при получении от процесса-получателя подтверждения о начале приёма отправленного сообщения, при этом отправленное сообщение или полностью принято процессом-получателем или находится в состоянии приёма.
19. Приём сообщений при помощи функции `MPI_Recv` может быть выполнен:
- a) от любого адресата и с любым тегом при указании специальных значений в качестве параметров вызова функции,
  - b) от любого адресата, однако, тег сообщения должен быть указан однозначно,
  - c) только от однозначно определяемого адресата с заданным тегом.
20. Прием сообщения при помощи функции `MPI_Recv` может быть инициирован:
- a) только до момента начала отправки сообщения;
  - b) только после момента начала отправки сообщения;
  - c) до момента, в момент или после момента начала отправки сообщения;
  - d) только в момент начала отправки сообщения.
21. Все данные для передачи в качестве сообщения MPI описываются с помощью триады:
- a) адрес памяти, ранг процесса-отправителя, используемый коммуникатор,
  - b) адрес памяти, ранг процесса-получателя, используемый коммуникатор.
  - c) адрес памяти, количество и тип элементов данных,
22. Функция `MPI_Recv`:
- a) в зависимости от используемой операции передачи может как заблокировать, так и не заблокировать процесс-получатель.
  - b) блокирует процесс-получатель до момента фактического получения сообщения,
  - c) принимает сообщение в фоновом режиме, процесс в это время может продолжать вычисления.

23. Указание используемого коммуникатора является:
- а) обязательным для всех операций передачи данных в MPI,
  - б) необязательным для некоторых операций передачи данных в MPI,
  - в) обязательным для некоторых операций передачи данных в MPI.
24. Под коммуникатором в MPI понимается:
- а) группа процессов, в рамках которой выполняются операции передачи данных,
  - б) пара процессов, в рамках которой происходит информационное взаимодействие.
  - в) специально создаваемый служебный объект, объединяющий в своем составе группу процессов и ряд дополнительных параметров, используемых при выполнении операций передачи данных,
25. Под параллельной программой в рамках MPI понимается:
- а) множество одновременно работающих процессоров.
  - б) множество одновременно выполняемых процессов,
  - в) множество одновременно выполняемых потоков,
26. Процессы параллельной программы в рамках MPI:
- а) могут выполняться на разных процессорах, на одном процессоре могут располагаться несколько процессов,
  - б) обязательно выполняются на одном процессоре.
  - в) могут выполняться только на разных процессорах,
27. Завершение вызова функции неблокирующего обмена означает:
- а) фактическое выполнение обмена данными;
  - б) фактическое выполнение приема данных (для функции неблокирующего приема) или начало фоновой передачи (для функции неблокирующей передачи);
  - в) инициацию запрошенной операции передачи, но ничего не говорит о начале или завершённости обмена.
28. Дана матрица размером  $N \times M$ . Написать программу транспонирования матрицы. Предложите способ разбиения задачи на подзадачи, опишите информационные связи между подзадачами.
29. На рисунке изображён граф информационных зависимостей подзадач некоторой задачи. Каким образом можно укрупнить подзадачи? Каким будет граф после укрупнения?



30. Даны матрицы размером  $N \times M$  и  $M \times N$ . Написать программу умножения матриц. Предложите способ разбиения задачи на подзадачи, опишите информационные связи между подзадачами.
31. Как передать с 1-го процесса на 8-й целое число 5 в рамках коммуникатора MPI\_COMM\_WORLD?
32. Как передать с 3-го процесса на 0-й массив вещественных чисел из 10 элементов в рамках коммуникатора MPI\_COMM\_WORLD?
33. Как передать с 0-го процесса всем остальным массив из 20 символов в рамках коммуникатора MPI\_COMM\_WORLD?

34. На нулевом процессе сформирован массив из 100 целых чисел. Как переслать равные части массива на все процессы коммуникатора MPI\_COMM\_WORLD, если в коммуникатор входит 10 процессов?
35. На 10 процессах коммуникатора MPI\_COMM\_WORLD вычислены промежуточные значения. Напишите функцию, которая перемножит данные с 10-ти процессов и передаст результат на 0-й процесс.
36. На 15 процессах коммуникатора MPI\_COMM\_WORLD вычислены промежуточные значения. Напишите функцию, которая найдёт максимальное значение и передаст результат на 0-й процесс.
37. Даны два вектора. Нужно найти их скалярное произведение. Предложите способ разбиения данной задачи на подзадачи, которые могут выполняться параллельно. Опишите характер связей между предложенными подзадачами.

**Разработчики:**

\_\_\_\_\_ доцент, к.п.н. Сокольская М.А.

Программа составлена в соответствии с требованиями ФГОС ВО по направлению подготовки 02.04.02 «Фундаментальная информатика и информационные технологии» (уровень магистратуры), утвержденный приказом Министерства образования и науки Российской Федерации от «23» августа 2017 г. № 811, зарегистрированный в Минюсте России «13» сентября 2017 г. № 48168 с изменениями и дополнениями от 26 ноября 2020 г., 8 февраля 2021 г.

Программа рассмотрена на заседании кафедры Алгебраических и информационных систем ИМИТ ИГУ «24» марта 2022 г.

Протокол № 9 Зав. кафедрой \_\_\_\_\_ Пантелеев В.И.

*Настоящая программа, не может быть воспроизведена ни в какой форме без предварительного письменного разрешения кафедры-разработчика программы.*