Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Баламирзоев Назим Лиодинович
Должность: Ректор
Дата подписания: 25.03.2026 16:00:30
Уникальный программный ключ:
5cf0d6f89e80f49a334f6a4ba58e91f3326b9926

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
Институт кибербезопасности и цифровых технологий

Региональный партнёр
ФГБОУ ВО
«Дагестанский государственный технический университет»

УТВЕРЖДАЮ
И.о. ректора ФГБОУ ВО
_____ Н.Л. Баламирзоев
«___» 09 2023 г.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Б1.В.ДВ.02.01 Высокопроизводительные вычисления

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Направленность (профиль подготовки): «Прикладной искусственный интеллект»

Квалификация выпускника бакалавр

Форма обучения очная

Махачкала 2023

1. ПАСПОРТ
фонда оценочных средств

по дисциплине Б1.В.ДВ.02.01 Высокопроизводительные вычисления

1.1. Результаты обучения по дисциплине:

Код	Наименование компетенции	Индикатор достижения компетенции	В результате освоения дисциплины обучающийся должен:	Другая дисциплина (дисциплины) / практика, участвующая в формировании компетенции
ОПК -4	Способен разрабатывать компоненты аппаратно-программных комплексов и баз данных, используя современные инструментальные средства и технологии программирования	ОПК-4.1 Применяет технологии параллельного программирования (OpenMP, MPI) для разработки компонентов вычислительных систем. ные системы. |	| Знать: архитектурные модели параллельных систем (SMID, MIMD), основы технологий OpenMP и MPI. Уметь: разрабатывать параллельные программы для систем с разделяемой и распределенной памятью. Владеть: навыками отладки и оптимизации параллельных приложений. Архитектура ЭВМ и систем. Программирование. Операцион	Архитектура ЭВМ и систем. Программирование. Операционные системы.
		ОПК-4.2 Использует технологии GPU-программирования (CUDA/OpenCL) для решения вычислительных задач.	Знать: архитектуру GPU, модель выполнения CUDA, иерархию памяти. Уметь: разрабатывать и оптимизировать ядра (kernels) для выполнения на графических процессорах. Владеть: инструментами разработки и профилирования CUDA-приложений. Ал-	Алгоритмы и структуры данных. Математические основы искусственного интеллекта.

			горитмы и а.	
ПК-1	Способен проводить предпроектное обследование, анализ предметной области, проектировать и моделировать компоненты информационных систем	ПК-1.1 Анализирует вычислительную задачу на предмет распараллеливания и выбирает адекватную HPC-технологию.	Знать: метрики производительности (ускорение, эффективность), законы Амдала и Густафсона. Уметь: проводить анализ алгоритма для выявления независимых ветвей вычислений и потенциальных узких мест. Владеть: методами проектирования параллельных алгоритмов.	Методы оптимизации. Теория алгоритмов.
		ПК-1.2 Проектирует архитектуру параллельного приложения для решения задач искусственного интеллекта.	знать: особенности распараллеливания алгоритмов машинного обучения (матричные операции, градиентный спуск). Уметь: проектировать гибридные (MPI+OpenMP) и распределенные решения для обучения моделей ИИ. Владеть: основами работы с системами управления заданиями (SLURM). | Системы	Системы искусственного интеллекта. Нейронные сети.

ПК-9	Способен инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем	ПК-9.1 Настраивает программную среду для разработки и выполнения параллельных приложений.	Знать: состав и требования к программному стеку HPC (компиляторы, библиотеки, драйверы). Уметь: устанавливать и настраивать MPI-среду (OpenMPI), CUDA Toolkit, библиотеки для HPC. Владеть: навыками работы в среде Linux, использования контейнеризации (Docker/Singularity) для HPC.	Администрирование информационных систем.
		ПК-9.2 Оценивает и подбирает аппаратные ресурсы для выполнения высокопроизводительных вычислений. |	Знать: характеристики современных процессоров, GPU, сетевых технологий (InfiniBand). Уметь: оценивать требуемые вычислительные ресурсы (ядра, память, дисковое пространство) для задачи. Владеть: основами чтения и интерпретации спецификаций аппаратного обеспечения.	Архитектура ЭВМ и систем.

1.2. Программа оценивания контролируемой компетенции:

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции / индикатора	Наименование оценочного средства
1.	Введение в HPC. Архитектуры параллельных систем.	ОПК-4/ОПК-4.1 ПК-1/ПК-1.1	Собеседование. Контрольный тест (Блок А).
2.	Параллельное программирование на разделяемой памяти (OpenMP).	ОПК-4/ОПК-4.1 ПК-1/ПК-1.1	Защита лабораторной работы. Контрольный тест

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции / индикатора	Наименование оценочного средства
			(Блок Б).
3.	Параллельное программирование на распределенной памяти (MPI).	ОПК-4/ОПК-4.1 ПК-9/ПК-9.1	Защита лабораторной работы. Контрольный тест (Блок В).
4.	Гибридное программирование (MPI + OpenMP).	ПК-1/ПК-1.2 ПК-9/ПК-9.1	Защита лабораторной работы. Контрольный тест (Блок Г).
5.	Вычисления на графических процессорах (CUDA).	ОПК-4/ОПК-4.2 ПК-9/ПК-9.2	Защита лабораторной работы. Контрольный тест (Блок Г).
6.	HPC для задач искусственного интеллекта. Оптимизация и профилирование.	ПК-1/ПК-1.2 ОПК-4/ОПК-4.2	Собеседование. Контрольный тест (Блок Д). Отчет по СРС.

3. Контрольные вопросы и задания

3.1. Вопросы к зачету по дисциплине **Б1.В.ДВ.02.01 Высокопроизводительные вычисления**

1. Предмет и цели высокопроизводительных вычислений (HPC). Сферы применения.
2. Классификация Флинна: SISD, SIMD, MISD, MIMD. Примеры современных архитектур.
3. Основные архитектуры параллельных систем: многопроцессорные системы с общей памятью, кластеры, гриды, облачные HPC-сервисы.
4. Понятия ускорения (speedup), эффективности (efficiency) и масштабируемости (scalability).
5. Закон Амдала: формулировка, ограничения, практический смысл.
6. Закон Густафсона: формулировка, отличие от закона Амдала.
7. Модель программирования OpenMP. Основные директивы, клаузы, функции среды выполнения.
8. Распараллеливание циклов в OpenMP. Клаузы `private`, `shared`, `reduction`.
9. Синхронизация в OpenMP: директива `barrier`, критическая секция (`critical`), атомарные операции (`atomic`).
10. Модель задач (Tasks) в OpenMP. Директива `task`. Области применения.
11. Модель программирования MPI. Понятия коммуникатора, ранга, размера.
12. Точечные операции в MPI: `MPI_Send`, `MPI_Recv`. Блокирующий и неблокирующий режим.
13. Коллективные операции в MPI: `MPI_Bcast`, `MPI_Scatter`, `MPI_Gather`, `MPI_Reduce`.
14. Принципы гибридного программирования (MPI + OpenMP). Преимущества и сложности.
15. Архитектура графических процессоров (GPU) NVIDIA. Иерархия потоков: грид, блок, нить.
16. Модель памяти в CUDA: глобальная, разделяемая, локальная, константная, текстурная память.

17. Основы разработки CUDA-ядер (kernels). Синтаксис, ограничения.
18. Оптимизация передачи данных между хостом (CPU) и устройством (GPU).
19. Стратегии распараллеливания алгоритмов машинного обучения: Data Parallelism, Model Parallelism.
20. Обзор инструментов профилирования HPC-приложений: Intel VTune, NVIDIA Nsight, Scalasca.
21. Системы управления заданиями в кластерных средах (SLURM). Основные команды и скрипты.
22. Контейнеризация в HPC: Docker, Singularity/Apptainer. Цели использования.
23. Методы отладки параллельных приложений: логирование, использование отладчиков (GDB с поддержкой MPI).
24. Понятие «узкого места» (bottleneck) в параллельных вычислениях и методы его поиска.

3.2. Вопросы для проверки уровня обученности ЗНАТЬ и УМЕТЬ

1. Чем отличается параллельная система от распределенной?
2. Какой закон (Амдала или Густафсона) более применим для задач с большим объемом независимых данных?
3. Что означает директива `#pragma omp parallel for`?
4. Как в OpenMP организовать редукционную операцию для вычисления суммы элементов массива?
5. В чем разница между клаузами `firstprivate` и `lastprivate`?
6. Что такое коммуникатор `MPI_COMM_WORLD`?
7. Как с помощью коллективных операций MPI выполнить поэлементное сложение векторов, распределенных по процессам?
8. Почему при гибридном программировании на узле кластера обычно запускают один MPI-процесс на несколько OpenMP-потоков?
9. Сколько нитей будет запущено, если размер блока в CUDA задан как `dim3 blockDim(16, 16)`?
10. Какую функцию CUDA API используют для копирования данных с хоста на устройство?
11. Какой тип памяти GPU является самым быстрым, но с наименьшим объемом?
12. В чем основная идея алгоритма MapReduce и как он связан с HPC?
13. Что такое ROCm и с какой целью он разработан?
14. Как с помощью SLURM зарезервировать для задачи 2 узла по 8 ядер в каждом?
15. Какие преимущества дает использование контейнеров в научных вычислениях?

3.3. Задания для проверки обученности ВЛАДЕТЬ

Задание 1. Анализ масштабируемости (OpenMP)
Используя код, представленный ниже, исследуйте масштабируемость параллельной реализации вычисления числа π методом Монте-Карло. Постройте график зависимости ускорения от количества потоков (1, 2, 4, 8). Объясните полученные результаты с точки зрения закона Амдала.

```c
#include <stdio.h>
#include <stdlib.h>
#include <omp.h>
int main(int argc, char argv[]) {
   long long num_points = 100000000;
```

```
    long long inside_circle = 0;
    #pragma omp parallel
    {
        unsigned int seed = omp_get_thread_num();
        long long local_inside = 0;
        #pragma omp for
        for (long long i = 0; i < num_points; i++) {
            double x = (double)rand_r(&seed) / RAND_MAX;
            double y = (double)rand_r(&seed) / RAND_MAX;
            if (xx + yy <= 1.0) local_inside++;
        }
        #pragma omp atomic
        inside_circle += local_inside;
    }
    double pi = 4.0  inside_circle / (double)num_points;
    printf("PI = %f\n", pi);
    return 0;
}
```

Указания к выполнению: Скомпилируйте код с флагами `-fopenmp`. Измерьте время выполнения для разного числа потоков (переменная окружения `OMP_NUM_THREADS`). Рассчитайте ускорение как `T(1) / T(N)`. Укажите, какая часть алгоритма является последовательной.

Задание 2. Распределенная сортировка (MPI)

Реализуйте алгоритм сортировки слиянием с использованием MPI. Исходный массив генерируется на процессе с рангом 0 и равномерно разбивается между всеми процессами. Каждый процесс сортирует свою часть локально (например, с помощью `qsort`). Затем процессы попарно обмениваются данными и выполняют слияние отсортированных частей. Сравните время выполнения с последовательной версией на 2 и 4 процессах.

Указания к выполнению: Используйте операции `MPI_Scatter` для распределения данных и `MPI_Gather` для сбора результатов. Для попарного слияния можно использовать схему, аналогичную битонической сортировке.

Задание 3. Гибридное умножение матриц (OpenMP + MPI)

Разработайте гибридную программу умножения плотных матриц. Используйте MPI для распределения блоков матрицы между узлами кластера и OpenMP для распараллеливания вычислений внутри каждого узла (на многоядерном CPU). Оцените эффективность гибридного подхода по сравнению с чистой MPI-реализацией.

Указания к выполнению: Реализуйте алгоритм Фокса или алгоритм Кэннона для распределения данных между MPI-процессами. Внутри каждого процесса используйте директивы OpenMP для распараллеливания циклов вычисления элементов результирующего блока.

Задание 4. Реализация CUDA-ядра для поэлементной операции

Напишите и оптимизируйте CUDA-ядро для вычисления значений функции `sigmoid` для всех элементов одномерного массива большой размерности (например, 10^7 элементов). Сравните производительность с CPU-реализацией (однопоточной и многопоточной с OpenMP). Объясните факторы, влияющие на ускорение.

Указания к выполнению: Используйте шаблон `__global__ void sigmoid_kernel(float in, float out, int n)`. Обратите внимание на выравнивание доступа к памяти и организацию циклов. Измеряйте время выполнения на GPU с помощью `cudaEvent`.

Задание 5. Профилирование и оптимизация
Для предоставленного преподавателем кода гибридного приложения (MPI+OpenMP), решающего задачу теплопроводности, выполните:
1. Профилирование с помощью инструмента `perf` или `Intel VTune` для выявления узких мест (hotspots, cache-misses).
2. Внесите изменения в код на основе результатов профилирования (например, измените схему разбиения данных, оптимизируйте циклы).
3. Оцените изменение производительности после оптимизаций.
Указания к выполнению: Представьте отчет, включающий графики из профилировщика до и после оптимизации, объяснение предпринятых шагов и количественную оценку улучшения (ускорение, снижение нагрузки на кэш).

3.4. Контрольные тесты (30 заданий)

КОНТРОЛЬНЫЕ ТЕСТЫ
по дисциплине «Высокопроизводительные вычисления»

Тест 1.
Вопрос: К какому классу по классификации Флинна относится современный многоядерный CPU с общей памятью, выполняющий разные потоки инструкций над разными данными?
Варианты ответов:
A) SISD
B) SIMD
C) MISD
D) MIMD
Верно: D) MIMD

Тест 2.
Вопрос: Основная причина ограничения ускорения (speedup) параллельной программы, описываемая законом Амдала, — это...
Варианты ответов:
A) Задержки при передаче данных по сети
B) Наличие последовательной (не распараллеливаемой) части программы
C) Неравномерность загрузки вычислительных ядер
D) Ограниченная пропускная способность памяти
Верно: B) Наличие последовательной (не распараллеливаемой) части программы

Тест 3.
Вопрос: Закон Густафсона акцентирует внимание на возможности увеличения...
Варианты ответов:
A) Эффективности использования процессорного времени
B) Объема решаемой задачи при увеличении числа процессоров
C) Тактовой частоты каждого процессора
D) Размера кэш-памяти
Верно: B) Объема решаемой задачи при увеличении числа процессоров

Тест 4.
Вопрос: TOP500 – это рейтинг, ранжирующий суперкомпьютерные системы прежде всего по производительности в тесте...
Варианты ответов:
A) LINPACK (HPL)

B) SPECfp
C) HPCG
D) Graph500
Верно: A) LINPACK (HPL)

Тест 5.
Вопрос: Индекс энергоэффективности суперкомпьютеров (Green500) измеряется в...
Варианты ответов:
A) Флопсах на ватт (FLOPS/W)
B) Джоулях на операцию (J/FLOP)
C) Ваттах на терафлопс (W/TFLOPS)
D) Килограммах эквивалента CO2
Верно: A) Флопсах на ватт (FLOPS/W)

Тест 6.
Вопрос: Директива `#pragma omp parallel` в OpenMP создает...
Варианты ответов:
A) Один дополнительный поток для выполнения следующего за ней цикла
B) Группу (team) потоков, которые выполняют заключенную в блок структурированную часть кода
C) Критическую секцию для синхронизации потоков
D) Задачу (task) для асинхронного выполнения
Верно: B) Группу (team) потоков, которые выполняют заключенную в блок структурированную часть кода

Тест 7.
Вопрос: Для распределения итераций цикла между потоками с помощью планировщика `dynamic` в OpenMP используется клауза...
Варианты ответов:
A) `schedule(static)`
B) `schedule(dynamic)`
C) `schedule(runtime)`
D) `schedule(guided)`
Верно: B) `schedule(dynamic)`

Тест 8.
Вопрос: Клауза `reduction` в OpenMP предназначена для...
Варианты ответов:
A) Уменьшения числа потоков в параллельной области
B) Сокращения объема используемой памяти
C) Выполнения редукционной операции (например, суммирования) над переменной из разных потоков
D) Оптимизации доступа к кэш-памяти
Верно: C) Выполнения редукционной операции (например, суммирования) над переменной из разных потоков

Тест 9.
Вопрос: Какая из перечисленных директив OpenMP используется для явной синхронизации всех потоков группы в барьерной точке?
Варианты ответов:
A) `#pragma omp critical`
B) `#pragma omp barrier`

C) `#pragma omp atomic`
D) `#pragma omp single`
Верно: B) `#pragma omp barrier`

Тест 10.
Вопрос: Укажите правильный формат директивы для распараллеливания цикла `for` с редукционной операцией суммы для переменной `sum`.
Варианты ответов:
A) `#pragma omp for sum(+)`
B) `#pragma omp for reduction(+:sum)`
C) `#pragma omp parallel for reduction(sum)`
D) `#pragma omp parallel reduction(+:sum)`
Верно: B) `#pragma omp for reduction(+:sum)`

Тест 11.
Вопрос: Директива `#pragma omp task` позволяет реализовать модель параллелизма, наиболее подходящую для...
Варианты ответов:
A) Регулярных вычислений в плотных циклах
B) Неравномерных, рекурсивных или асинхронных задач
C) Только для вложенных циклов
D) Синхронного обмена сообщениями
Верно: B) Неравномерных, рекурсивных или асинхронных задач

Тест 12.
Вопрос: Функция `MPI_Init` должна быть вызвана...
Варианты ответов:
A) Только в главном процессе (ранг 0)
B) В каждом процессе параллельного приложения, до любых других вызовов MPI
C) После создания всех коммуникаторов
D) Только если используется более одного процесса
Верно: B) В каждом процессе параллельного приложения, до любых других вызовов MPI

Тест 13.
Вопрос: Для передачи массива данных от одного процесса (корня) всем другим процессам в коммуникаторе используется операция...
Варианты ответов:
A) `MPI_Scatter`
B) `MPI_Bcast` (Broadcast)
C) `MPI_Gather`
D) `MPI_Reduce`
Верно: B) `MPI_Bcast` (Broadcast)

Тест 14.
Вопрос: Функция `MPI_Send` в режиме по умолчанию является...
Варианты ответов:
A) Блокирующей (ожидает, пока система не примет данные для отправки)
B) Неблокирующей (возвращает управление немедленно)
C) Синхронной (ожидает начала соответствующего приема)
D) Буферизованной (всегда использует внутренний буфер)
Верно: A) Блокирующей (ожидает, пока система не примет данные для отправки)

Тест 15.
Вопрос: Операция `MPI_Reduce` выполняет...
Варианты ответов:
A) Рассылку данных от всех процессов одному
B) Сбор данных от всех процессов к одному
C) Глобальную редукцию (например, суммирование) данных со всех процессов с записью результата в один процесс
D) Попарный обмен данными между процессами
Верно: C) Глобальную редукцию (например, суммирование) данных со всех процессов с записью результата в один процесс

Тест 16.
Вопрос: Коммуникатор `MPI_COMM_WORLD` включает в себя...
Варианты ответов:
A) Только процессы на одном физическом узле
B) Все процессы, запущенные в рамках одного параллельного задания (job)
C) Процессы, созданные с помощью функции `MPI_Comm_split`
D) Процессы, участвующие в коллективной операции
Верно: B) Все процессы, запущенные в рамках одного параллельного задания (job)

Тест 17.
Вопрос: Для безопасного попарного обмена данными между процессами (каждый процесс одновременно отправляет и принимает) следует использовать функции...
Варианты ответов:
A) `MPI_Send` и `MPI_Recv` в произвольном порядке
B) `MPI_Sendrecv`
C) `MPI_Isend` и `MPI_Irecv`
D) `MPI_Bcast`
Верно: B) `MPI_Sendrecv`

Тест 18.
Вопрос: Функция `MPI_Scatter` распределяет данные...
Варианты ответов:
A) От одного процесса (корня) всем процессам, разбивая отправляемый массив на равные части
B) От всех процессов одному, собирая массив из частей
C) От одного процесса всем, отправляя каждому один и тот же массив
D) Между всеми процессами попарно
Верно: A) От одного процесса (корня) всем процессам, разбивая отправляемый массив на равные части

Тест 19.
Вопрос: Гибридная модель программирования MPI+OpenMP обычно применяется для...
Варианты ответов:
A) Упрощения отладки приложений
B) Более эффективного использования иерархии памяти на кластерах с многоядерными узлами
C) Запуска приложений на GPU
D) Реализации только задач типа «очередь задач» (task queue)
Верно: B) Более эффективного использования иерархии памяти на кластерах с многоядерными узлами

Тест 20.
Вопрос: В архитектуре CUDA блоки потоков (thread blocks) выполняются на...
Варианты ответов:
A) Разных CPU в кластере
B) Один потоковый мультипроцессор (Streaming Multiprocessor, SM) GPU
C) В отдельном кэше L1
D) В глобальной памяти GPU
Верно: B) Один потоковый мультипроцессор (Streaming Multiprocessor, SM) GPU

Тест 21.
Вопрос: Тип памяти GPU с наибольшей скоростью доступа, но наименьшим объемом (порядка десятков килобайт на блок) – это...
Варианты ответов:
A) Глобальная память (global memory)
B) Локальная память (local memory)
C) Разделяемая память (shared memory)
D) Константная память (constant memory)
Верно: C) Разделяемая память (shared memory)

Тест 22.
Вопрос: Для копирования данных с хоста (CPU) на устройство (GPU) в CUDA используется функция...
Варианты ответов:
A) `cudaMemcpyToSymbol`
B) `cudaMemcpy` с параметром `cudaMemcpyHostToDevice`
C) `cudaMemcpy` с параметром `cudaMemcpyDeviceToHost`
D) `cudaMalloc`
Верно: B) `cudaMemcpy` с параметром `cudaMemcpyHostToDevice`

Тест 23.
Вопрос: Иерархия параллелизма в CUDA от большего к меньшему: Грид (Grid) -> ? -> ? -> Нить (Thread).
Варианты ответов:
A) Варп (Warp), Блок (Block)
B) Блок (Block), Варп (Warp)
C) Мулытипроцессор (SM), Блок (Block)
D) Сегмент (Segment), Блок (Block)
Верно: B) Блок (Block), Варп (Warp)

Тест 24.
Вопрос: Основное преимущество гибридного подхода (MPI+OpenMP) перед чистым MPI на кластере с многоядерными узлами – это...
Варианты ответов:
A) Упрощение логики программирования
B) Снижение накладных расходов на межпроцессное взаимодействие (MPI) за счет обмена данными через общую память внутри узла (OpenMP)
C) Возможность использования GPU
D) Автоматическая балансировка нагрузки
Верно: B) Снижение накладных расходов на межпроцессное взаимодействие (MPI) за счет обмена данными через общую память внутри узла (OpenMP)

Тест 25.

Вопрос: Пропускная способность (bandwidth) памяти GPU является ключевым фактором для производительности задач, которые...
Варианты ответов:
A) Требуют интенсивных тригонометрических вычислений
B) Являются ограниченными пропускной способностью памяти (memory-bound)
C) Содержат много условных переходов
D) Используют только регистры
Верно: B) Являются ограниченными пропускной способностью памяти (memory-bound)

Тест 26.
Вопрос: Распараллеливание обучения нейронной сети методом Data Parallelism предполагает...
Варианты ответов:
A) Разделение слоев сети между разными устройствами
B) Разделение мини-батча обучающих данных между устройствами с последующей синхронизацией градиентов
C) Последовательное обучение разных частей модели
D) Использование только CPU для разных задач
Верно: B) Разделение мини-батча обучающих данных между устройствами с последующей синхронизацией градиентов

Тест 27.
Вопрос: Фреймворк Horovod от Uber используется для...
Варианты ответов:
A) Визуализации нейронных сетей
B) Упрощения реализации распределенного обучения (data parallelism) поверх различных backends (TensorFlow, PyTorch)
C) Автоматического синтеза архитектур нейронных сетей
D) Сжатия обученных моделей
Верно: B) Упрощения реализации распределенного обучения (data parallelism) поверх различных backends (TensorFlow, PyTorch)

Тест 28.
Вопрос: Основная вычислительная операция при обучении полносвязных сетей, эффективно распараллеливаемая на GPU – это...
Варианты ответов:
A) Поэлементное сложение векторов
B) Матричное умножение (GEMM)
C) Вычисление функции активации
D) Операция if-else
Верно: B) Матричное умножение (GEMM)

Тест 29.
Вопрос: Параметр `world_size` в контексте распределенного обучения нейронных сетей означает...
Варианты ответов:
A) Размер одного батча данных
B) Общее количество процессов (например, GPU), участвующих в обучении
C) Количество слоев в нейронной сети
D) Размер словаря в NLP-задачах
Верно: B) Общее количество процессов (например, GPU), участвующих в обучении

Тест 30.
Вопрос: Стратегия Model Parallelism (распараллеливание модели) применяется, когда...
Варианты ответов:
A) Очень большой объем данных
B) Модель слишком велика, чтобы поместиться в память одного устройства (GPU)
C) Требуется максимальная скорость на одном GPU
D) Используется только CPU-кластер
Верно: B) Модель слишком велика, чтобы поместиться в память одного устройства (GPU)

Тест 31.
Вопрос: Какой инструмент НЕ является типичным профилировщиком для HPC-приложений?
Варианты ответов:
A) Intel VTune Profiler
B) NVIDIA Nsight Systems
C) Scalasca
D) Apache Kafka
Верно: D) Apache Kafka

Тест 32.
Вопрос: Директива OpenMP `#pragma omp single` указывает, что...
Варианты ответов:
A) Цикл должен выполняться одним потоком
B) Заключенный в блок код должен быть выполнен только одним (любым) потоком группы
C) Переменная должна быть приватной для одного потока
D) Создается одна задача
Верно: B) Заключенный в блок код должен быть выполнен только одним (любым) потоком группы

Тест 33.
Вопрос: Для чего в CUDA используется атрибут `__global__` при объявлении функции?
Варианты ответов:
A) Для функций, выполняемых только на CPU
B) Для функций (ядер), вызываемых с хоста и выполняемых на GPU
C) Для функций, выполняемых на GPU и вызываемых только из других GPU-функций
D) Для глобальных переменных в памяти GPU
Верно: B) Для функций (ядер), вызываемых с хоста и выполняемых на GPU

Тест 34.
Вопрос: Система управления пакетными заданиями SLURM используется в HPC-кластерах прежде всего для...
Варианты ответов:
A) Компиляции параллельных программ
B) Планирования, выделения ресурсов и запуска вычислительных задач (jobs)
C) Мониторинга температуры процессоров
D) Организации распределенной файловой системы
Верно: B) Планирования, выделения ресурсов и запуска вычислительных задач (jobs)

Тест 35.
Вопрос: Контейнеризация (например, с помощью Singularity/Apptainer) в HPC-средах решает проблему...
Варианты ответов:
A) Низкой тактовой частоты CPU

B) Воспроизводимости вычислений и зависимости от конкретного системного окружения
C) Отсутствия компиляторов Fortran
D) Малого объема оперативной памяти
Верно: B) Воспроизводимости вычислений и зависимости от конкретного системного окружения

Критерии оценивания теста:

«Отлично» (5): 85-100% правильных ответов (26-30 баллов).
«Хорошо» (4): 70-84% правильных ответов (21-25 баллов).
«Удовлетворительно» (3): 51-69% правильных ответов (16-20 баллов).
«Неудовлетворительно» (2): менее 50% правильных ответов (0-15 баллов).

5. Описание показателей и критериев оценивания с указанием шкалы оценивания для очной формы обучения (с применением балльно-рейтинговой системы):

Оценка по дисциплине складывается из текущего рейтинга и рейтинга на зачете.

Текущий рейтинг складывается из оценок, полученных при защите лабораторных работ. Для лабораторных работ 1–16 максимальное число баллов за работу равно 3. Для работ 17–18 максимальное число баллов за работу равно 6. Студент считается сдавшим лабораторные работы, если он сдал все работы и набрал не менее 36 баллов.

Рейтинг на зачете определяется следующим образом:

Ответы на первый и второй вопросы — до 10 баллов, выполнение задания — до 10 баллов, дополнительные вопросы в рамках курса до 10 баллов.

Оценивание ответов на первый, второй и дополнительные вопросы:

9-10 баллов выставляется, если студент демонстрирует полное понимание вопросов, правильность ответов, полное и логически последовательное изложение материала.

7-8 баллов выставляется, если студент демонстрирует: значительное понимание вопросов, правильность, но недостаточную полноту ответов на заданные теоретические вопросы; допущение неточности ответа*;*

5-6 баллов выставляется, если студент демонстрирует: понимание вопросов, по существу излагает материал, но не усвоил его деталей, есть погрешности в ответах; допускает неточности, недостаточно правильные формулировки, нарушения логической последовательности в изложении материала;

Менее 5 баллов выставляется, если студент демонстрирует: непонимание вопросов; студент не знает значительной части материала, не ответил на дополнительные вопросы или отказался от ответов на вопросы и задания.

Оценивание задания:

9-10 баллов выставляется, если студент демонстрирует полное понимание заданий, правильность ответов; полное, точное и логически последовательное изложение материала;

7-8 баллов выставляется, если студент демонстрирует: значительное понимание заданий, правильность, но недостаточную полноту ответов на заданные задания; допущение неточности ответа*;*

5-6 баллов выставляется, если студент демонстрирует: понимание заданий, основные этапы задания выполнены, но есть погрешности в ответах

Менее 5 баллов выставляется, если студент демонстрирует: непонимание заданий; основные шаги задания не выполнены или выполнены неправильно, не ответил на дополнительные вопросы или отказался от ответов на вопросы и задания.

Минимальный балл рейтинга на зачете в соответствии с положением о рейтинге равен 24, максимальный балл — 40.

В итоге по курсу, суммируя итоги текущего рейтинга и рейтинга на зачете оценка "зачтено" выставляется обучающемуся, если он набрал не менее 60 баллов.

Сведения о дополнениях и изменениях, внесенных в ФОС дисциплины

Учебный год	Решение кафедры (№ протокола, дата)	Внесенные в ФОС дополнения и изменения	Подпись заведующего кафедрой