



Федеральное агентство по рыболовству  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Калининградский государственный технический университет»  
(ФГБОУ ВО «КГТУ»)

УТВЕРЖДАЮ  
И.о. директора института

Фонд оценочных средств  
(приложение к рабочей программе модуля)  
**«МАССОВО-ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ (GPU)»**

основной профессиональной образовательной программы бакалавриата  
по направлению подготовки  
**09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА**

ИНСТИТУТ  
РАЗРАБОТЧИК

цифровых технологий  
кафедра прикладной информатики

## 1 РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ, ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ И КРИТЕРИИ ОЦЕНИВАНИЯ

### 1.1 Результаты освоения дисциплины

Таблица 1 – Планируемые результаты обучения по дисциплине, соотнесенные с установленными компетенциями

Код и наименование компетенции	Дисциплина	Результаты обучения (владения, умения и знания), соотнесенные с компетенциями
ПК-1 «Способен формулировать требования, проектировать и разрабатывать программное обеспечение на языках высокого уровня»	Массово-параллельные вычисления (GPU)	<p><u>знать:</u> основные подходы проектирования компьютерного программного обеспечения; анализ требований к программному обеспечению; основные подходы организации исследовательских и проектных работ; правила оформления отчетов о проведении научно-исследовательской работе и подготовки публикаций по результатам исследования;</p> <p><u>уметь:</u> применять методы алгоритмизации, языки и технологии программирования, для создания и использования продуктов и услуг; разрабатывать технические спецификации на программные компоненты и их взаимодействие; применять основные подходы организации исследовательских и проектных работ; применять правила оформления отчетов о проведении научно-исследовательской работе и подготовки публикаций по результатам исследования;</p> <p><u>владеть:</u> способами управления процессами создания и использования продуктов и</p>

		услуг в сфере информационно-коммуникационных технологий, в том числе навыками разработки алгоритмов и программ для их практической реализации; навыки проектирования программного обеспечения; навыками организации исследовательских и проектных работ; навыками оформления отчетов о проведении научно-исследовательской работе и подготовки публикаций по результатам исследования;
--	--	---

1.2 К оценочным средствам текущего контроля успеваемости относятся:

- тестовые задания открытого и закрытого типов.

К оценочным средствам для промежуточной аттестации относятся:

- типовые задания по курсовому проекту;
- типовые задания по контрольной работе для заочной формы обучения.

1.3 Критерии оценки результатов освоения дисциплины

Универсальная система оценивания результатов обучения включает в себя системы оценок: 1) «отлично», «хорошо», «удовлетворительно», «неудовлетворительно»; 2) «зачтено», «не зачтено»; 3) 100 – балльную/процентную систему и правило перевода оценок в пятибалльную систему (табл. 2).

Таблица 2 – Система оценок и критерии выставления оценки

Система оценок	2	3	4	5
	0-40%	41-60%	61-80 %	81-100 %
Критерий	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	«не зачтено»	«зачтено»		
<b>1 Системность и полнота знаний в отношении изучаемых объектов</b>	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полнотой знаний и системным взглядом на изучаемый объект

Система оценок	2	3	4	5
	0-40%	41-60%	61-80 %	81-100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
Критерий	«не зачтено»	«зачтено»		
	некоторые из которых может связывать между собой)			
<b>2 Работа с информацией</b>	Не в состоянии находить необходимую информацию, либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпретировать и систематизировать необходимую информацию в рамках поставленной задачи	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
<b>3 Научное осмысление изучаемого явления, процесса, объекта</b>	Не может делать научно корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно корректный анализ предоставленной информации, вовлекает в исследование новые релевантные задаче данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные поставленной задаче данные, предлагает новые ракурсы поставленной задачи
<b>4 Освоение стандартных алгоритмов решения профессиональных задач</b>	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

Система оценок	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
	<b>0-40%</b>	<b>41-60%</b>	<b>61-80 %</b>	<b>81-100 %</b>
Критерий	<b>«неудовлетворительно»</b>	<b>«удовлетворительно»</b>	<b>«хорошо»</b>	<b>«отлично»</b>
	<b>«не зачтено»</b>	<b>«зачтено»</b>		
	алгоритм, допускает ошибки		предложенного алгоритма	

## 2 ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

ПК-1 «Способен формулировать требования, проектировать и разрабатывать программное обеспечение на языках высокого уровня»

### Тестовые задания закрытого типа:

1. Библиотека MPI предназначена для работы на следующих типах многопроцессорных вычислительных систем

**1. С распределенной памятью**

2. С общей памятью
3. с отсутствием памяти
4. с оперативной памятью

2. Библиотека OpenMP предназначена для работы на следующих типах многопроцессорных вычислительных систем

**1. С распределенной памятью**

**2. С общей памятью**

3. с отсутствием памяти
4. с оперативной памятью

3. Можно выполнять вычисления на основе CUDA на графических ускорителях производителя

**1. AMD**

**2. Nvidia**

3. Intel
4. Каких угодно

4. Отметьте все варианты типов функций для отправки сообщений в MPI.

**1. Парные**

**2. Коллективные**

3. Выборочные
4. Отдельные

5. Функция MP\_Barrier выполняет следующее действие:

**1. Задержка отправки сообщения**

**2. Синхронизация всех процессов**

3. Проверка сообщений
4. Проверка работы системы

6. Количество потоков в OpenMP-приложении связано с количеством ядер процессора, на котором это приложение запускается, следующим образом:

1. Количество потоков должно быть равно количеству ядер процессора
2. Количество потоков не должно превышать количество ядер процессора

**3. Никак не связано**

7. Система очередей на кластере/суперкомпьютере предназначена?

1. Для регулирования доступа пользовательских задач к ресурсам кластера
2. Для регулирования доступа пользователей на кластер в порядке приоритета.

**3. Все перечисленное**

### **Тестовые задания открытого типа:**

8. Напишите фрагмент кода, который выводит номер MPI-процесса и число MPI-процессов.

**Ответ** `int size,rank;`

```
    MPI_Comm_size(MPI_COMM_WORLD,&size);
    MPI_Comm_rank(MPI_COMM_WORLD,&rank);
    printf("#%d of %d\n",rank,size);
```

9. Ниже приведен фрагмент кода, который с помощью MPI суммирует поэлементно массивы, находящиеся в памяти разных процессов. Вставьте название пропущенной функции.

```
double x[N],y[N];
_____ (x,y,N,MPI_DOUBLE, MPI_SUM,0,comm);
```

**Ответ: MPI\_Reduce**

9. Сложение значений переменной по всем потокам библиотеки OpenMP выполняет директива \_\_\_\_\_ .

**Ответ: omp reduction**

10. Ниже приведен фрагмент кода, который с помощью MPI вычисляет скалярное произведение двух векторов

```
double s=0.0, sum;
for (i=0;i<N;i++) s += x[i]*y[i];
MPI_Reduce(&s,&sum,1,MPI_DOUBLE,_____,0,MPI_COMM_WORLD);
```

Вместо \_\_\_\_\_ должен быть код операции

**Ответ: MPI\_SUM**

11. Ниже приведен фрагмент кода, который с помощью MPI выполняет пересылку целочисленного массива длины 5 от процесса номер 0 процессу номер 1.

```
#define N 5
```

```
double x[N];
if (rank==0)
    MPI_Send(x,N,_____,2,123,comm);
if (rank==2)
    MPI_Recv(x,N,_____,0,123,comm,MPI_STATUS_IGNORE);
```

На месте \_\_\_\_\_ должна быть константа для типа данных.

**Ответ: MPI\_DOUBLE.**

12. В приведенном фрагменте кода на месте \_\_\_\_\_ необходимо поставить функции, которые возвращают количество MPI-процессов и номер MPI-процесса.

```
_____ (MPI_COMM_WORLD,&size);
_____ (MPI_COMM_WORLD,&rank);
```

**Ответ: MPI\_Comm\_size, MPI\_Comm\_rank**

13. В приведенном ниже фрагменте кода на месте пропуска поставьте функцию, которая выводит количество потоков OpenMP

```
#pragma omp parallel
{
    int n;
    n = _____ ();
    printf("number of threads %d\n",n);
}
```

**Ответ: omp\_get\_num\_threads**

14. В приведенном ниже фрагменте кода на месте пропуска поставьте функцию, которая выводит номер потока OpenMP и количество потоков OpenMP

```
#pragma omp parallel
{
    int i;
    i = _____ ();
    printf("I'm thread %d \n",i);
}
```

**Ответ: omp\_get\_thread\_num**

15. Директива private в OpenMP означает \_\_\_\_\_

**Ответ: создание экземпляра переменной, упомянутой в директиве private отдельного для каждого потока.**

16. В приведенном ниже фрагменте кода на месте пропуска поставьте директиву OpenMP, необходимую для распараллеливания цикла по итерациям.

```
#define N 100
#pragma omp _____ \ reduction(+:s)
for (i=0;i< N;i++)
```

**Ответ: omp parallel for**

17. В приведенном ниже фрагменте кода на месте пропуска поставьте название заголовочного файла, необходимого для компиляции и исполнения данного кода

```
#define N 10000
#include <_____>
#include <math.h>
int i,s = 0;
double f,x;
#pragma omp parallel for \ reduction(+:s)
for (i=0;i< N;i++)
{
    x = i*M_PI/2/N;
    f = x*x;
    s += f;
}
s /= N;
```

**Ответ: omp.h**

18. Определите порядок распределения нагрузки между потоками OpenMP в каждом из трех вариантов: static, guided, dynamic

**Ответ: static – статическое распределение, dynamic – динамическое распределение, guided – управляемое распределение**

19. Директива \_\_\_\_\_ обеспечивает возможность одномоментного доступа к переменной только одного потока?

**Ответ: OpenMP atomic**

20. Фрагмент кода выполняется только одним потоком OpenMP благодаря директивам \_\_\_\_\_?

**Ответ: 1) single и 2) critical section**

21. Следующие языки программирования \_\_\_\_\_ позволяют работать с OpenMP

**Ответ: C/C++, Fortran**

22. Графические ускорители производителя \_\_\_\_\_ позволяют работать с технологией CUDA

**Ответ: Nvidia**

23. К типу данных \_\_\_\_\_ принадлежат стандартные глобальные переменные CUDA

**Ответ: dim3**

24. Поточный блок CUDA – это \_\_\_\_\_?

**Ответ: группа потоков, между которыми возможна синхронизация, и которые могут иметь общий доступ к разделяемой памяти.**

25. Вывод результата на консоль непосредственно из программы, выполняемой на графическом ускорителе \_\_\_\_\_

**Ответ: невозможен.**

26. Порядок вывода на консоль результат, полученный на графическом ускорителе с помощью CUDA следующий \_\_\_\_\_

**Ответ: сначала скопировать переменную, содержащую результат из памяти графического ускорителя в оперативную память компьютера с помощью функции `cudaMemcpy`, затем вывести на консоль с помощью обычных средств ввода-вывода.**

27. Максимальное количество потоков в сетке (grid) потоковых блоков в CUDA определяется \_\_\_\_\_

**Ответ: параметрами графического ускорителя и версией CUDA**

28. Технология CUDA дает преимущества по сравнению с OpenMP и MPI для задач \_\_\_\_\_

**Ответ: с преобладанием простых однотипных вычислений**

29. Константу \_\_\_\_\_ нужно задать в качестве параметра функции `cudaMemcpy` для копирования данных из оперативной памяти компьютера в память графического ускорителя

**Ответ: `cudaMemcpyHostToDevice`**

30. Эквивалент в CUDA директивы `omp atomic` - это

**Ответ: атомарные операции**

### **3 ТИПОВЫЕ ЗАДАНИЯ НА КОНТРОЛЬНУЮ РАБОТУ, КУРСОВУЮ РАБОТУ/КУРСОВОЙ ПРОЕКТ, РАСЧЕТНО-ГРАФИЧЕСКУЮ РАБОТУ**

Типовые задания по курсовому проекту представлены ниже. Необходимо представить программный пример реализации задания.

1. Степени параллелизма. Статическое и динамическое распараллеливание последовательных программ
2. Распараллеливание ациклических участков
3. Особенности распараллеливание выражений
4. Задача распараллеливания выражений
5. Распараллеливание циклических фрагментов программ

6. Алгоритм репликации для управления распределенной памятью
7. Алгоритм полного размножения для управления распределенной памятью
8. Операционные системы параллельных вычислительных систем. Планирование процессов
9. Задача оптимального отображения параллельных процессов на архитектуру многопроцессорной вычислительной системы
10. Постановка задачи оптимального отображения
11. Балансировка загрузки
12. Статическая и динамическая балансировка загрузки
13. Динамическая балансировка загрузки
14. Операционные системы мультипроцессоров. Планирование процессов
15. Планирование процессов в мультипроцессоре
16. Планирование независимых процессов
17. Планирование зависимых процессов
18. Операционные системы мультикомпьютеров. Планирование процессов
19. Балансировка загрузки, инициируемая отправителем
20. Балансировка загрузки, инициируемая получателем
21. Иерархический графовый алгоритм балансировки загрузки
22. Рекурсивное огрубление графа на основе паросочетаний
23. Рекурсивное огрубление графа на основе паросочетаний из тяжелых клик
24. Рекурсивная бисекция графа
25. Языки высокого уровня для программирования векторно-конвейерных и векторнопараллельных вычислительных систем

Темы контрольных работ для заочной формы обучения представлены ниже.

1. Пространство решений
2. Параллельные алгоритмы
3. Практичные системы программирования
4. Экспериментальные верификаторы
5. Модели параллелизма в языках программирования
6. Языки сверхвысокого уровня
7. Параллельное программирование. APL
8. Теоретико-множественное программирование
9. Высокопроизводительное программирование.

10. Параллельное функциональное программирование.
11. Многопоточность и многопроцессность
12. Трансформационная семантика
13. Абстрактный комплекс
14. Реализационная прагматика
15. Настраиваемая кодогенерация
16. Память
17. Нормальные формы
18. Распределенные программные системы

**4 СВЕДЕНИЯ О ФОНДЕ ОЦЕНОЧНЫХ СРЕДСТВ И ЕГО СОГЛАСОВАНИИ**

Фонд оценочных средств для аттестации по дисциплине «Массово-параллельные вычисления (GPU)» представляет собой компонент основной профессиональной образовательной программы бакалавриата по направлению подготовки 09.03.03 «Прикладная информатика».

Преподаватель-разработчик – д.т.н., профессор А. В. Снытников

Фонд оценочных средств рассмотрен и одобрен заведующим кафедрой прикладной информатики.

Заведующий кафедрой



М.В. Соловей

Фонд оценочных средств рассмотрен и одобрен методической комиссией института цифровых технологий (протокол №5 от 29.08.2024 г).

Председатель методической комиссии



О.С. Витренко