

«

»

“ ”

“ ”
_____ .

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ Параллельное программирование

: 09.03.04

, :

: 3, : 6

,

		6
1	()	3
2		108
3	, .	61
4	, .	18
5	, .	0
6	, .	36
7	, .	8
8	, .	2
9	, .	5
10	, .	47
11	(, ,)	
12		

(): 09.03.04

229 12.03.2015 ., : 01.04.2015 .

: 1,

(): 09.03.04

, 6 20.06.2017

, 6 21.06.2017

:

,

:

,

:

. . .

1.

1.1

Компетенция ФГОС: ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов; в части следующих результатов обучения:	
1.	
10.	
Компетенция ФГОС: ПК.1 готовность применять основные методы и инструменты разработки программного обеспечения; в части следующих результатов обучения:	
7.	,
8.	- ,

2.

2.1

(, , ,)	
-----------	--

.1. 7 ,	
1. об общей совокупности проблем, возникающих при использовании параллельных высокопроизводительных вычислительных систем для решения научных и производственных задач	;
.1. 8 - ,	
2. о достигнутом на современном этапе уровне результатов теоретических исследований в области параллельного программирования	;
.3. 1	
3. об уровне практической реализации теоретических результатов для автоматизации распараллеливания алгоритмов	;
.1. 8 - ,	
4. об актуальных вопросах теории параллельных вычислений	;
.3. 1	
5. объект курса (параллельные высокопроизводительные вычислительные системы), предмет курса (параллельные алгоритмы), задачи курса (методы распараллеливания и средства организации выполнения параллельных программ), место теории параллельных вычислений в общей совокупности изучаемых дисциплин	;
.1. 7 ,	
6. архитектуру высокопроизводительных вычислительных систем, типовые топологии схем коммутации, классификацию многопроцессорных вычислительных систем	;
7. модели параллельных вычислительных процессов, концепцию неограниченного параллелизма, модели многопроцессорных систем с общей и распределенной памятью, модель конвейерной системы	;
8. модель алгоритма в виде графа "операнд - операции", представление алгоритма в виде графа потока данных, расписание параллельных вычислений	;

9.модель параллельных вычислений в виде сети Петри, основные проблемы параллельных вычислений: синхронизация, взаимоисключение, блокировка (тупики)		;
10.потокową модель параллельных вычислений в виде графа "процесс-ресурс", понятие процесса, проблемы взаимодействия процессов, синхронизация параллельных процессов, аппарат событий		;
.3. 1		
11.параллелизм данных и параллелизм задач, показатель эффективности распараллеливания (ускорение), эффективность использования вычислительной системы, способы оценки показателей эффективности		; ;
.1. 7		
12.уровни распараллеливания вычислений, распараллеливание вычислений на уровне команд, выражений, программных модулей, отдельно выполняемых заданий		; ;
13.параллельные языки программирования и расширения стандартных языков, средства автоматического распараллеливания, параллельные компиляторы, параллельные предметные библиотеки, инструментальные системы для проектирования параллельных программ		;
14.общие принципы построения и реализации MPI, общие функций MPI, коммуникаторы, функции обмена сообщениями типа "точка-точка": блокирующий и неблокирующий обмен, синхронные и стандартные послышки сообщений		; ;
15.технологии программирования OpenMP, последовательные и параллельные нити программы, организацию параллельных секций, параллельные циклы, директивы синхронизации, спецификации OpenMP для языков C и C++.		; ;
16.систему разработки параллельных программ CUDA, мобильность и эффективность выполнения программ, состав системы CUDA, основные директивы распараллеливания, использование отладчика и анализатора производительности CUDA-программ.		; ;
17.перспективы развития многопроцессорных ВС и параллельного программирования.		;
.3. 10		
18.разрабатывать, отлаживать и запускать на исполнение параллельные программы с использованием функций MPI		;
19.разрабатывать, отлаживать и запускать на исполнение параллельные программы с использованием функций OpenMP		;
20.разрабатывать, отлаживать и запускать на исполнение параллельные программы с использованием функций CUDA		;
21.выполнять анализ эффективности распараллеливания алгоритмов с использованием различных систем организации взаимодействия ветвей		;
22.использования компиляторов и библиотек, поддерживающих параллельное программирование		;
23.запуска, отладки, трассировки, анализа лог-файлов параллельных программ в однородных и гетерогенных вычислительных комплексах		;

3.

3.1

		,	.		
: 6					
:					

1.	.	0	1	1, 4	.
:					
2.	.	0	1	5, 6	.
:					
3.	.	0	2,5	10, 7, 8, 9	.
:					
5.	.	0	2	11, 12, 13, 3	- . , . . , . .

6. OpenMP.	0	2	13, 15, 5	OpenMP. C ++.
7. CUDA -	0	2	11, 16, 5	CUDA.
8. MPI	0	7	11, 14, 5, 6	MPI. MPL, " - ": : :
9.	0	0,5	17, 2, 3, 5	

: 6				
:				
1. OpenMP.	0	4	15, 19, 22, 23	()
2. OpenMP.	2	4	11, 15, 19, 23	OpenMP ,
3. CUDA.	0	4	12, 16, 20, 22	NVidia CUDA
4. OpenMP CUDA	2	4	19, 20, 22, 23	OpenMP CUDA
5. MPI.	0	4	14, 18, 22	MPI.
6. MPI.	2	4	14, 18, 22, 23	MPI: ,
7. MPI.	2	4	14, 18, 21, 23	MPI- MPE
8.	0	8	18, 19, 20, 21, 22, 23	OpenMP, CUDA MPI

4.

: 6				
1		18, 19, 20, 21, 22, 23	16	0
<p>... []: - : / . . . ; . . . - . - , [2015]. - : http://elibrary.nstu.ru/source?bib_id=vtls000214351. - : " []: []: - :] / . . . ; . . . - . - , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html. - :</p>				
2		1, 2, 3, 4, 5, 6	18	5
<p>: . . . " []: []: - : , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html. - .</p>				
3		7, 8, 9	8	0
<p>3 : " " / . . . - ; [. . . , . . .]. - , 2015. - 30, [2] . : .. - : http://elibrary.nstu.ru/source?bib_id=vtls000215108</p>				
4		10, 11, 12, 13, 14, 15, 16, 17, 9	5	0
<p>" : . . . " []: - / . . . ; . . . - . - , [2011]. - : http://ciu.nstu.ru/fulltext/unofficial/2011/lib_874_1321433132.doc. - : - . - , 2008. - 310, [1] . :</p>				

5.

- , (. 5.1).

5.1

	-
	: http://vt.cs.nstu.ru/~malyavko/ParProgr
	e-mail:translab@ngs.ru
	e-mail:translab@ngs.ru
	: http://vt.cs.nstu.ru/~malyavko/ParProgr

1	:
Краткое описание применения: Подготовка материалов для расчетно-графической работы путем сбора результатов трассировки всех разрабатываемых в лабораторных работах параллельных программ.	
" [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."	

6.

(), - 15- ECTS.
. 6.1.

6.1

	.	
: 6		
Лабораторная №1:	2	4
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №2:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №3:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №4:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №5:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №6:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №7:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
Лабораторная №8:	4	8
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		
РГЗ:	0	20
- " [] : [] / . ; , [2010]. - : http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html . - ."		

Зачет:	0	20
-		

6.2

6.2

.3	1.		+
	10.	+	
.1	7.		+
	8.		+

1

7.

1. Малявко А. А. Параллельное программирование на основе технологий OpenMP, MPI, CUDA : учебное пособие / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, 2015. - 114, [1] с. : табл. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000215088
 2. Малявко А. А. Параллельное программирование [Электронный ресурс] : электронный учебно-методический комплекс / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2015]. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000214351. - Загл. с экрана.
 3. Малышкин В. Э. Параллельное программирование мультимикропроцессоров : [учебник] / В. Э. Малышкин, В. Д. Корнеев. - Новосибирск, 2011. - 295 с. : ил., табл.
 4. Гергель В. П. Высокопроизводительные вычисления для многопроцессорных многоядерных систем : [учебник для вузов по направлениям ВПО 010400 "Прикладная математика и информатика" и 010300 "Фундаментальная информатика и информационные технологии"] / В. П. Гергель ; Б-ка Нижегород. гос. ун-та им. Н. И. Лобачевского. - М., 2010. - 539, [4] с. : ил., схемы, табл. - На обл. в вых. дан.: Суперкомпьютерный консорциум университетов России.
 5. Воеводин В. В. Параллельные вычисления : [учебник для вузов по направлению 510200 "Прикладная математика и информатика"] / В. В. Воеводин, Вл. В. Воеводин. - СПб., 2004. - 599 с. : ил.
1. Корнеев В. В. Вычислительные системы / В. В. Корнеев. - М., 2004. - 510, [1] с. : ил.
 2. Малышкин В. Э. Основы параллельных вычислений. Ч. 1 : Учеб. пособие для IV курса фак. прикладной математики и информатики / Новосиб. гос. техн. ун-т. - Новосибирск, 1998. - 60 с.
 3. Малышкин В. Э. Основы параллельных вычислений. Ч. 2 : [учебное пособие для 4 курса ФПМИ] / В. Э. Малышкин ; Новосиб. гос. техн. ун-т. - Новосибирск, 2000. - 51 с. : ил.
 4. Немнюгин С. А. Параллельное программирование для многопроцессорных вычислительных систем / Сергей Немнюгин, Ольга Стесик. - СПб., 2002. - 396 с. : ил.
 5. Эндрюс Г. Д. Основы многопоточного, параллельного и распределенного программирования : [пер. с англ.] / Грегори Р. Эндрюс. - М., 2003. - 505 с. : ил.

6. Макконелл Д. Основы современных алгоритмов : учебное пособие по направлению "Информатика и вычислительная техника" / Дж. Макконелл ; пер. с англ. под ред. С. К. Ландо ; доп. М. В. Ульянова. - М., 2006. - 366 с. : ил.
7. Корнеев В. Д. Параллельное программирование в MPI / Отв. ред. В. Э. Малышкин. - Новосибирск, 2002. - 214 с. : ил.
8. Богачев К. Ю. Основы параллельного программирования / К. Ю. Богачёв. - М., 2003. - 342 с.

1. ЭБС НГТУ : <http://elibrary.nstu.ru/>
2. ЭБС «Издательство Лань» : <https://e.lanbook.com/>
3. ЭБС IPRbooks : <http://www.iprbookshop.ru/>
4. ЭБС "Znanium.com" : <http://znanium.com/>
5. :

8.

8.1

1. Малявко А. А. Сайт дисциплины "Параллельное программирование" [Электронный ресурс] : [электронный учебно-методический комплекс] / А. А. Малявко ; Новосиб. гос. техн. ун-т. - Новосибирск, [2010]. - Режим доступа: <http://vt.cs.nstu.ru/~malyavko/ParProgr/index.html>. - Загл. с экрана.
2. Параллельное программирование для систем с общей памятью : методические указания к лабораторным работам по дисциплине "Параллельное программирование" для 3 курса дневного отделения АВТФ направления "Информатика и вычислительная техника" / Новосиб. гос. техн. ун-т ; [сост.: В. В. Ландовский, Е. Н. Павенко]. - Новосибирск, 2015. - 30, [2] с. : ил.. - Режим доступа: http://elibrary.nstu.ru/source?bib_id=vtls000215108
3. Корнеев В. Д. Методические указания для лабораторных по курсу "Основы параллельного программирования" [Электронный ресурс] : учебно-методическое пособие / В. Д. Корнеев ; Новосиб. гос. техн. ун-т. - Новосибирск, [2011]. - Режим доступа: http://ciu.nstu.ru/fulltext/unofficial/2011/lib_874_1321433132.doc. - Загл. с экрана.
4. Корнеев В. Д. Параллельное программирование кластеров : учебное пособие / В. Д. Корнеев ; Новосиб. гос. техн. ун-т. - Новосибирск, 2008. - 310, [1] с. : ил.

8.2

- 1 Visual Studio 2010
- 2 Система программирования CUDA
- 3 MPICH-2.0

9.

1	(Internet
	Internet)	

Новосибирск 2017

1. Обобщенная структура фонда оценочных средств учебной дисциплины

Обобщенная структура фонда оценочных средств по дисциплине Параллельное программирование приведена в Таблице.

Таблица

Формируемые компетенции	Показатели сформированности компетенций (знания, умения, навыки)	Темы	Этапы оценки компетенций	
			Мероприятия текущего контроля (курсовой проект, РГЗ(Р) и др.)	Промежуточная аттестация (экзамен, зачет)
ОПК.3 готовность применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов	з1. знать зависимость между сложностью и временем выполнения программного проекта	CUDA - система разработки параллельных программ для графического процессора. Заключение. Обзор современного состояния средств и технологий параллельного программирования. Интерфейс передачи сообщений MPI Технология программирования OpenMP.		Зачет, вопросы 1-36
ОПК.3	у10. уметь распараллеливать алгоритмы и программы для решения прикладных задач	Основы программирования на CUDA. Совместное использование всех технологий. Совместное использование технологий OpenMP и CUDA Технология MPI. Основы. Технология MPI. Расширенные средства. Технология MPI. Средства трассировки и отладки параллельных приложений. Технология программирования OpenMP. Основы. Технология программирования OpenMP. Расширенные средства.	РГЗ, разделы 1.1, 1.2, 1.3	
ПК.1/ПТ готовность применять основные методы и инструменты разработки программного обеспечения	з7. знать современные компиляторы, отладчики и оптимизаторы программного кода	CUDA - система разработки параллельных программ для графического процессора. Введение в предмет курса. Обзор средств и методов организации параллельных вычислений. Заключение. Обзор современного состояния средств и технологий параллельного программирования. Интерфейс передачи сообщений MPI Технология программирования OpenMP.		Зачет, вопросы 37-67
ПК.1/ПТ	з8. знать компоненты программно-технических архитектур, существующие приложения и интерфейсы взаимодействия с ними	Введение в предмет курса. Обзор средств и методов организации параллельных вычислений. Заключение. Обзор современного состояния средств и технологий параллельного программирования.		Зачет, вопросы 68-88

2. Методика оценки этапов формирования компетенций в рамках дисциплины.

Промежуточная аттестация по дисциплине проводится в 6 семестре - в форме дифференцированного зачета, который направлен на оценку сформированности компетенций ОПК.3, ПК.1/ПТ.

Зачет проводится в устной форме, по билетам.

Кроме того, сформированность компетенций проверяется при проведении мероприятий текущего контроля, указанных в таблице раздела 1.

В 6 семестре обязательным этапом текущей аттестации является расчетно-графическое задание (работа) (РГЗ(Р)). Требования к выполнению РГЗ(Р), состав и правила оценки сформулированы в паспорте РГЗ(Р).

Общие правила выставления оценки по дисциплине определяются балльно-рейтинговой системой, приведенной в рабочей программе дисциплины.

На основании приведенных далее критериев можно сделать общий вывод о сформированности компетенций ОПК.3, ПК.1/ПТ, за которые отвечает дисциплина, на разных уровнях.

Общая характеристика уровней освоения компетенций.

Ниже порогового. Уровень выполнения работ не отвечает большинству основных требований, теоретическое содержание курса освоено частично, пробелы могут носить существенный характер, необходимые практические навыки работы с освоенным материалом сформированы не достаточно, большинство предусмотренных программой обучения учебных заданий не выполнены или выполнены с существенными ошибками.

Пороговый. Уровень выполнения работ отвечает большинству основных требований, теоретическое содержание курса освоено частично, но пробелы не носят существенного характера, необходимые практические навыки работы с освоенным материалом в основном сформированы, большинство предусмотренных программой обучения учебных заданий выполнено, некоторые виды заданий выполнены с ошибками.

Базовый. Уровень выполнения работ отвечает всем основным требованиям, теоретическое содержание курса освоено полностью, без пробелов, некоторые практические навыки работы с освоенным материалом сформированы недостаточно, все предусмотренные программой обучения учебные задания выполнены, качество выполнения ни одного из них не оценено минимальным числом баллов, некоторые из выполненных заданий, возможно, содержат ошибки.

Продвинутый. Уровень выполнения работ отвечает всем требованиям, теоретическое содержание курса освоено полностью, без пробелов, необходимые практические навыки работы с освоенным материалом сформированы, все предусмотренные программой обучения учебные задания выполнены, качество их выполнения оценено числом баллов, близким к максимальному.

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

Паспорт зачета

по дисциплине «Параллельное программирование», 6 семестр

1. Методика оценки

Зачет проводится в устной форме, по билетам. Билет формируется по следующему правилу: первый вопрос выбирается из диапазона вопросов 1-35, второй вопрос из диапазона вопросов 36- 67, третий вопрос из диапазона вопросов 68-88 (список вопросов приведен ниже). В ходе зачета преподаватель вправе задавать студенту дополнительные вопросы из общего перечня (п. 4).

Форма билета для зачета

НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
Факультет АВТФ

Билет № _____

к зачету по дисциплине «Параллельное программирование»

1. Каким ограничениям должен удовлетворять цикл для того, чтобы к нему можно было применить директиву for OpenMP?
2. С помощью какого расширения синтаксиса запускается ядро CUDA?
3. Что такое виртуальная топология в MPI? Как она связана с физической топологией вычислительной сети/кластера/комплекса?

Утверждаю: зав. кафедрой _____ должность, ФИО
(подпись)

(дата)

2. Критерии оценки

- Ответ на билет для зачета считается **неудовлетворительным**, если студент при ответе на вопросы не дает определений основных понятий, не способен показать причинно-следственные связи явлений, при решении задачи допускает принципиальные ошибки, оценка составляет 0-9 *баллов*.
- Ответ на билет для зачета засчитывается на **пороговом** уровне, если студент при ответе на вопросы дает определение основных понятий, может показать причинно-следственные связи явлений, при решении задачи допускает непринципиальные ошибки, например, вычислительные, оценка составляет 10-14 *баллов*.
- Ответ на билет для зачета билет засчитывается на **базовом** уровне, если студент при

ответе на вопросы формулирует основные понятия, законы, дает характеристику процессов, явлений, проводит анализ причин, условий, может представить качественные характеристики процессов, не допускает ошибок при решении задачи, оценка составляет 15-18 баллов.

- Ответ на билет для зачета билет засчитывается на **продвинутом** уровне, если студент при ответе на вопросы проводит сравнительный анализ подходов, проводит комплексный анализ, выявляет проблемы, предлагает механизмы решения, способен представить количественные характеристики определенных процессов, приводит конкретные примеры из практики, не допускает ошибок и способен обосновать выбор метода решения задачи, оценка составляет 19-20 баллов.

3. Шкала оценки

Зачет считается сданным, если сумма баллов по всем вопросам билета оставляет не менее 10 баллов (из 20 возможных).

В общей оценке по дисциплине баллы за зачет учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Вопросы к зачету по дисциплине «Параллельное программирование»

1. Перечислите составные части технологии OpenMP.
2. С помощью какой директивы (директив) создаются новые параллельные области программы?
3. Что такое критическая секция программы?
4. Каким образом можно установить нужное количество потоков для создания очередной параллельной области?
5. Как обеспечить выполнение фрагмента параллельной области только главным потоком?
6. Какая опция директивы OpenMP *for* используется для указания способа распределения итераций цикла между потоками?
7. Что такое *deadlock*? Каким правилам нужно следовать, чтобы избежать возможности попадания параллельной программы в *deadlock*?
8. Что такое сведение данных? Какие опции и в каких директивах используются для выполнения сведения?
9. Что делает директива OpenMP *threadprivate*?
10. Как обеспечить выполнение фрагмента параллельной области потоком с максимальным номером в данной параллельной области?
11. Для чего используется опция *firstprivate*? Чем она отличается от опций *private* и *lastprivate*?
12. Что такое вложенная параллельная область программы? В каких случаях ее нельзя создать?
13. Что такое неявная барьерная синхронизация? С помощью каких средств ее можно отменить?
14. Для чего используются директивы OpenMP *sections* и *section*? Что делает каждая из этих директив?
15. Перечислите все средства синхронизации потоков в OpenMP.
16. Перечислите возможные способы распределения итераций цикла между потоками.
17. Что делает директива OpenMP *atomic*?
18. В чем состоит различие между общими и локальными переменными потока?
19. С помощью каких средств можно ограничить глубину вложенности параллельных областей программы?

20. От чего зависит равномерность загрузки процессоров/ядер системы с общей памятью?
21. Каким образом функция, вызываемая из параллельной программы может выяснять, в последовательной или параллельной области она выполняется?
22. Как обеспечить выполнение фрагмента параллельной области в точности одним потоком?
23. Какое значение будет иметь переменная *count* в результате выполнения фрагмента параллельной программы:
24. `int count = 0;`
25. `#pragma omp parallel for`
26. `for(int i = 0; i < 10; i++) {`
27. `count++;`
28. `}`
29. Могут ли два потока одновременно захватить один множественный замок?
30. Для чего может использоваться опция *reduction* в некоторых директивах OpenMP?
31. Каково назначение директивы OpenMP *single*?
32. Каким ограничениям должен удовлетворять цикл для того, чтобы к нему можно было применить директиву OpenMP *for*?
33. Чем отличается простой замок от множественного?
34. Перечислите известные Вам механизмы синхронизации потоков в OpenMP.
35. Перечислите способы распределения итераций цикла между потоками параллельной области.
36. Что такое ядро в терминологии CUDA?
37. Как задается размерность и количество потоков ядра, выполняемого графическим процессором?
38. Перечислите виды и характеристики памяти, доступной из программы GPU.
39. Как указать требуемое размещение переменной в памяти GPU (регистровой, разделяемой, глобальной, памяти текстур, ...)?
40. Перечислите виды и характеристики памяти, доступной из программы CPU при использовании архитектуры CUDA.
41. Какова максимальная размерность блока в потоках?
42. Перечислите встроенные векторные типы данных расширения языка C и объясните смысл их наименований.
43. Как в программе для CPU обеспечить синхронизацию с программой для GPU?
44. Если к одному CPU подключено несколько видеокарт NVidia архитектуры CUDA, то каким образом можно обеспечить их одновременную загрузку?
45. Перечислите виды и характеристики памяти GPU, доступной из программы основного процессора.
46. Какие ограничения наложены на функции, которые должны выполняться на GPU?
47. Что такое мультипроцессор в терминологии CUDA?
48. Какие ограничения наложены на функции, выполняемые в графическом процессоре?
49. Как определить версию и технические характеристики графического процессора NVIDIA?
50. Перечислите и охарактеризуйте группы библиотечных функций Run-time библиотеки CUDA.
51. Что такое сетка, блок, поток в терминологии CUDA?
52. Что такое тип данных *dim3*?
53. Какова максимальная размерность сетки в блоках?
54. Перечислите и охарактеризуйте функции компонент программного обеспечения CUDA.
55. Можно ли (и если да – то как) в программе для CPU выяснить, какую версию спецификации CUDA реализует видеокарта?
56. Какую модель параллелизма реализует архитектура CUDA?

57. Что делает препроцессор nvcc?
58. Можно ли получить указатель на функцию, выполняемую графическим процессором?
59. С помощью какого расширения синтаксиса запускается ядро CUDA?
60. Перечислите основные группы функций CUDA и охарактеризуйте их назначение.
61. Перечислите и охарактеризуйте дополнительные виды функций, существующие в расширении синтаксиса языка C для CUDA.
62. Какие виды памяти доступны из программ для GPU?
63. Как синхронизируются процессы в CPU и в GPU?
64. Можно ли одновременно использовать технологии OpenMP и CUDA?
65. Какие имена полей используются во встроенных векторных типах?
66. С данными какого типа оперируют арифметико-логические устройства GPU CUDA?
67. Как узнать максимальное количество потоков в блоке для видеокарты NVIDIA, подключенной к CPU?
68. Какие виды виртуальных топологий можно реализовывать с использованием MPI?
69. Что такое барьерная синхронизация в MPI? Какие еще виды синхронизации существуют в стандарте MPI-1?
70. Перечислите основные группы функций стандарта MPI-1.
71. Может ли MPI-программа, выполняющаяся на многоядерном узле, использовать все его ядра и если да, то каким образом?
72. Что такое пользовательская операция редукции?
73. Чем различаются размер и протяженность типов данных MPI?
74. Что такое буферизованная передача сообщений?
75. Что такое коммуникатор? Какие проблемы решаются с использованием этого понятия?
76. Сколько в MPI операций приема сообщений типа «точка-точка»? Перечислите и охарактеризуйте их.
77. Что такое коллективное взаимодействие ветвей программы?
78. Как путем вызова одной функции можно одновременно передать и принять блок данных?
79. Перечислите предопределенные операции редукции данных MPI.
80. Что такое виртуальная топология в MPI? Как она связана с физической топологией вычислительной сети/кластера/комплекса?
81. Для чего в MPI-программах можно использовать производные типы данных?
82. Что понимается под операциями сдвига и циклического сдвига данных? Какие функции обычно используются для их реализации?
83. Что такое исключаящая редукция (MPI_Exscan), как она выполняется?
84. Может ли ветвь параллельной программы с помощью MPI передавать сообщения сама себе?
85. Какие операции рассылки и сбора данных реализованы в MPI?
86. Должны ли абсолютно все ветви параллельной программы принимать участие в коллективном взаимодействии?
87. Сколько конструкторов производных типов реализовано в MPI? Перечислите и охарактеризуйте их.
88. Чем различаются топологии «тор» и «решетка»?

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Новосибирский государственный технический университет»
Кафедра вычислительной техники

**Паспорт
расчетно-графического задания (работы)**

по дисциплине «Параллельное программирование», 6 семестр

1. Методика оценки

В рамках расчетно-графического задания (работы) по дисциплине студенты должны решить поставленную задачу, разработав последовательную и три варианта параллельной программы с использованием технологий OpenMP, CUDA/OpenCL и MPI.

При выполнении расчетно-графического задания (работы) студенты должны выполнить измерения затрат времени на решение задачи последовательной и каждой из трех вариантов параллельной программы при нескольких (от 4-х до 8-ми) вариантах значений исходных данных, проанализировать и обосновать получившиеся зависимости и соотношения, сформулировать выводы.

Обязательные структурные части РГЗ.

- a. Анализ поставленной задачи и разработка последовательного алгоритма ее решения.
- b. Выбор и обоснование алгоритмов распараллеливания для каждой из технологий поддержки параллельных вычислений: OpenMP, CUDA/OpenCL и MPI.
- c. Результаты вычислительных экспериментов, анализ зависимостей затрат времени от используемой технологии, значений исходных данных и масштабов распараллеливания. Выводы и заключение.
- d. Приложения: коды последовательной и параллельных программ.

Оцениваемые позиции: 1.1, 1.2, 1.3.

2. Критерии оценки

- Работа считается **не выполненной**, если выполнены не все части РГЗ(Р), отсутствует анализ поставленной задачи и все три параллельные программы, оценка составляет 0-9 баллов.
- Работа считается выполненной **на пороговом** уровне, если части РГЗ(Р) выполнены формально: анализ задачи не соответствует ее природе и технологии распараллеливания, разработаны только последовательная и одна из трех параллельных программ, анализ результатов отсутствует или выполнен по недостаточному для его проведения количеству вычислительных экспериментов, оценка составляет 10-12 баллов.
- Работа считается выполненной **на базовом** уровне, если анализ поставленной задачи выполнен на приемлемом для распараллеливания уровне, выполнено распараллеливание для двух технологий из трех, полученная степень ускорения решения задачи при распараллеливании соответствует ожидаемому хотя бы для одной технологии, оценка составляет 13-17 баллов.
- Работа считается выполненной **на продвинутом** уровне, если анализ задачи выполнен в полном объеме, разработаны и оптимизированы параллельные программы для всех трех технологий, вычислительные эксперименты проведены для широкого диапазона значений исходных данных и нескольких уровней распараллеливания, анализ результатов всесторонне обоснован, оценка составляет 18-20 баллов.

3. Шкала оценки

В общей оценке по дисциплине баллы за РГЗ(Р) учитываются в соответствии с правилами балльно-рейтинговой системы, приведенными в рабочей программе дисциплины.

4. Примерный перечень тем РГЗ(Р)

1. Поиск кратчайшего пути между заданными вершинами на заданном взвешенном графе
2. Определение количества различных путей между заданными вершинами на ориентированном графе
3. Определение количества различных непересекающихся по ребрам путей между заданными вершинами на ориентированном графе
4. Определение количества различных непересекающихся по вершинам путей между заданными вершинами на ориентированном графе
5. Поиск минимального охватывающего дерева взвешенного графа
6. Разбить заданный взвешенный неориентированный граф на заданное количество подграфов с минимальными суммами расстояний между вершинами в каждом подграфе
7. Найти минимальную правильную раскраску заданного графа
8. Решение системы линейных уравнений методом Гаусса
9. Решение системы линейных уравнений методом Якоби
10. Решение системы линейных уравнений методом Крамера
11. Решение системы линейных уравнений методом Гаусса-Зейделя
12. Решение системы линейных уравнений методом релаксации
13. Найти наименьшее число n , большее заданного N такое, что числа $2*n$, $3*n$, $4*n$, $5*n$ и $6*n$ содержат тот же набор цифр (цифр в них может быть больше, чем в числе n), что и число n
14. Для заданного положительного числа n найти количество различных способов его получения в результате суммирования заданного количества k положительных целых чисел (например, $5=1+1+3=1+2+2$ – итого 2 способа выразить 5 как сумму 3-х чисел).
15. Найти минимальное число, которое может быть разложено в сумму простых чисел не менее, чем N различными способами (например, для числа 10 существует ровно 5 таких способов: $10 = 7+3 = 5+5 = 2+3+5 = 2+2+3+3 = 2+2+2+2+2$)
16. Найти максимальное число, меньшее заданного, которое может быть представлено как сумма степеней 2, 3 и 4 простых чисел (минимальное такое число есть $28 = 2^2+2^3+2^4$)
17. Найти максимальное простое число, меньшее заданного N , десятичное представление которого содержит наибольшее количество одинаковых цифр.
18. Найти максимальное число, меньшее заданного, которое равно сумме его десятичных цифр, возведенной в степень, большую 1 (например, $512 = (5+1+2)^3$)
19. Сортировка пузырьковая (обменная)
20. Быстрая сортировка
21. Сортировка Шелла
22. Пирамидальная сортировка
23. Поразрядная сортировка
24. Вычисление определителя матрицы
25. Вычисление характеристического многочлена матрицы и его корней
26. Умножение матриц
27. Сложение матриц
28. Возведение матрицы в заданную степень
29. Умножение вектора на матрицу
30. Вычисление обратной матрицы
31. Найти компактную подматрицу заданного размера с максимальной суммой элементов в заданной матрице
32. Найти разреженную подматрицу заданного размера с максимальной суммой элементов в заданной матрице
33. Вычисление корней алгебраического уравнения одной переменной большой размерности на заданном интервале

34. Вычислить произведение двух заданных полиномов N переменных
35. Найти количество всех вхождений заданной подстроки в заданной строке (порядок следования символов важен, но символы подстроки в строке не обязательно должны следовать друг за другом, например, в строке $xxuzy$ подстрока xu встречается 4 раза, подстрока xz - 2 раза, подстроки xx , zy и yz - по одному разу, а подстроки ux , yy , zx и zz - ни разу)
36. Любое число n_0 , большее единицы, порождает последовательность вида:
- $$n_{i+1} = (n_i \% 2 == 0 ? n_i / 2 : 3 * n_i + 1),$$
- последним элементом которой является 1 (например: $13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$). Найти наибольшее, меньшее заданного N , число, порождающее самую длинную такую последовательность.
37. Число называется циркулярно простым, если любая круговая перестановка его цифр есть простое число (например – 197 или 31 или 17). Найти количество циркулярно простых чисел в диапазоне от N до M
38. Число называется укорачиваемым простым слева, если последовательное удаление его десятичных цифр, начиная со старшей, дает простые числа (пример: 3797, 797, 97, 7). Это же число является укорачиваемым простым справа (3797, 379, 37, 3). Найти максимальные укорачиваемые простые слева, справа, с обеих сторон числа, меньшие заданного N
39. Найти первые N последовательно возрастающих на величину k чисел, больших заданного числа, все делители которых различны (два таких числа, возрастающих на 1 – это $14=2*7$ и $15=3*5$)
40. Найти максимальное простое число, меньшее заданного N , которое может быть записано, как сумма последовательно возрастающих простых чисел
41. Найти наибольший палиндром (число, одинаково читающееся слева направо и справа налево), меньший заданного числа N , который является суммой квадратов последовательно расположенных чисел (например, число-палиндром $595 = 6^2 + 7^2 + 8^2 + 9^2 + 10^2 + 11^2 + 12^2$)
42. Найти наименьшее простое число, большее заданного N такое, что начало и окончание его десятичного представления являются последовательными простыми числами (например, число 3137 и числа 31 и 37)
43. Близнецами называются простые числа, разница между которыми равна двум (например: 29 и 31, 41 и 43, ...). Найти две соседние пары близнецов, расстояние между серединами которых больше заданного N
44. Найти наименьшие целые числа $x > y > z$, большие заданного N , такие, что $x+y$, $x-y$, $x+z$, $x-z$, $y+z$, $y-z$ являются полными квадратами других чисел
45. Найти наибольшее простое число, меньшее заданного N , такое, что последовательность его цифр в обратном порядке тоже является простым числом (такowymi являются, например, числа 31, 71, 101, ...)
46. Найти наименьшее число, большее заданного N , такое, что сумма квадратов его десятичных цифр есть полный квадрат (например, таким является число 442: $4^2 + 4^2 + 2^2 = 36 = 6^2$)
47. Найти наименьшее число n , большее заданного N , имеющее точно такое же количество различных делителей, какое имеет число $n+1$
48. Найти в заданном диапазоне от N_1 до N_2 ($N_2 > N_1$) два простых числа, имеющие минимальное и максимальное значения произведений начала и конца своего десятичного представления (учитывать все возможные способы разбиения этого представления на две части)
49. Найти минимальное число, большее заданного, которое равно сумме его десятичных цифр, возведенной в степень, большую 1 (например, $4913 = (4+9+1+3)^3$)
50. Найти минимальное число, большее заданного N , которое делится нацело на все заданные числа n_1, n_2, \dots, n_k

51. Найти наименьший палиндром (число, одинаково читающееся слева направо и справа налево), который является произведением двух различных n -значных чисел
52. Найти минимальное число, большее заданного, которое может быть представлено как сумма степеней 2, 3, 4 и 5 простых чисел (наименьшее такое число есть $60 = 2^2 + 2^3 + 2^4 + 2^5$)
53. Найти наименьшее простое число, большее заданного N , такое, что начало и окончание его десятичного представления являются последовательными простыми числами (например, число 3137 и числа 31 и 37)
54. Найти наименьшее простое число, большее заданного N , которое является суммой нескольких последовательно возрастающих простых чисел
55. Найти наименьшее простое число, большее заданного N , такое, что последовательность его цифр в обратном порядке тоже является простым числом (такowymi являются, например, числа 31, 71, 101, ...)
56. Найти максимальное число, меньшее заданного N , имеющее наибольшее количество не обязательно различных простых делителей
57. Найти минимальное число n , большее заданного, такое, что сумма его десятичных цифр равна сумме десятичных цифр числа $137 * n$
58. Существуют такие числа n , для которых сумма любого его делителя d с результатом деления n/d является простым числом (таково число 30, имеющее делители 1, 2, 3, 5, 6, 10, 15, 30). Найти наименьшее такое число, большее заданного N
59. Найти сумму всех целых чисел, меньших N , которые не делятся нацело на заданные числа n_1, n_2, \dots, n_k
60. Найти наименьшее целое число, которое делится без остатка на все числа в диапазоне $[n_1, n_2]$