

Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Институт прикладной математики и компьютерных наук

УТВЕРЖДЕНО:
Директор
А. В. Замятин

Оценочные материалы по дисциплине

Параллельное программирование

по направлению подготовки / специальности

10.05.01 Компьютерная безопасность

Направленность (профиль) подготовки/ специализация:
Анализ безопасности компьютерных систем

Форма обучения
Очная

Квалификация
Специалист по защите информации

Год приема
2024

СОГЛАСОВАНО:
Руководитель ОП
В.Н. Треньяев

Председатель УМК
С.П. Сущенко

Томск – 2024

1. Компетенции и индикаторы их достижения, проверяемые данными оценочными материалами

Целью освоения дисциплины является формирование следующих компетенций:

ОПК-7 Способен создавать программы на языках высокого и низкого уровня, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.

Результатами освоения дисциплины являются следующие индикаторы достижения компетенций:

ИОПК-7.1 Осуществляет построение алгоритма, проведение его анализа и реализации в современных программных комплексах

ИОПК-7.2 Понимает общие принципы построения и использования языков программирования высокого уровня и низкого уровня

ИОПК-7.3 Демонстрирует навыки создания программ с применением методов и инструментальных средств программирования для решения различных профессиональных, исследовательских и прикладных задач

ИОПК-7.4 Осуществляет обоснованный выбор инструментария программирования и способов организации программ

2. Оценочные материалы текущего контроля и критерии оценивания

Элементы текущего контроля:

– билеты для письменного тестирования;

Перечень теоретических вопросов для письменного тестирования: (ОПК-7)

Тема 1.

1. Дайте современную классификацию суперкомпьютеров.
2. Каких суперкомпьютеров больше: кластеров или MPP-машин?
3. Что такое тест Linpack? Каков его результат?
4. Что такое флопс?
5. Какая операционная система чаще используется на вычислительных кластерах?
6. Какая страна произвела больше всех самых производительных суперкомпьютеров в мире?
7. Какова производительность самого мощного суперкомпьютера в мире?
8. Как называется самый производительный суперкомпьютер в России?
9. Что такое top500? Сколько электроэнергии (Вт) потребляет самый мощный суперкомпьютер?
10. Какова производительность в флопсах кластера ТГУ Cyberia?
11. Самый мощный суперкомпьютер в мире имеет производительность: 1 Exaflops или 1 Petaflops? Дайте правильный ответ.
12. Нарисуйте архитектуру компьютера с общей памятью.
13. Нарисуйте архитектуру компьютера с распределенной памятью.
14. В каких единицах измеряется производительность компьютера?
15. Что такое степень параллелизма компьютера? Чему она равна для конкретных компьютеров?
16. Что такое степень параллелизма алгоритма?
17. Как определяется ускорение параллельной программы? В каких пределах оно может меняться?
18. Можно ли распространить определение ускорения для параллельного алгоритма? Дайте комментарий.
19. На каких математических принципах базируется алгоритм сдваивания? В чем его преимущество перед обычным алгоритмом суммирования?

20. Сформулируйте закон Амдала. Каким будет максимальное ускорение параллельной программы, в которой 1% арифметических операций выполняется только одним процессором?
21. Пользуясь формальной моделью ускорения, определите условия, при которых будет иметь место максимальный параллелизм, параллелизм среднего уровня и отсутствие ускорения параллельной программы.
22. Перечислите основные этапы разработки параллельной программы. Какие из них зависят от архитектуры ЭВМ, какие не зависят?
23. Каковы требования к проведению первого этапа построения параллельного алгоритма?
24. Чем отличается мелкозернистый параллелизм алгоритма от крупнозернистого?
25. В чем отличие функциональной декомпозиции от декомпозиции по данным?
26. Перечислите типы коммуникаций. Дайте им краткую характеристику.

Тема №2 Рекуррентные формулы

1. Приведите примеры использования рекуррентных соотношений в вычислительной математике.
2. В чем заключается проблема распараллеливания рекуррентных соотношений?
3. Что собой представляет диаграмма маршрутизации алгоритма и для чего она используется?
4. Опишите алгоритм каскадной схемы суммирования.
5. В чем заключается параллелизм каскадной схемы суммирования?
6. За счет чего удастся обеспечить равномерную работу ПЭ при выполнении каскадной схемы суммирования?
7. Опишите алгоритм сдваивания для вычисления суммы элементов последовательности.
8. Оцените количество операций сложения в алгоритме сдваивания при вычислении суммы $n=2^q$ чисел.
9. Опишите алгоритм модифицированной каскадной схемы суммирования элементов последовательности (МКСС).
10. Каким ограничениям должно соответствовать число элементов суммируемой последовательности в МКСС?
11. Дайте оценку ускорения и эффективности МКСС.
12. В чем заключается особенность МКСС по сравнению с алгоритмом сдваивания? За счет чего эта особенность проявляется?
13. В чем заключается основная идея метода циклической редукции?
14. Как применяется метод циклической редукции для распараллеливания линейной рекуррентной формулы первого порядка?
15. После применения циклической редукции к линейным рекуррентным формулам удастся ли получить явные формулы?
16. Какова степень параллелизма алгоритма циклической редукции?

Тема №3 Вопросы про MPI

1. Что означает термин «процесс» в MPI?
2. Как нумеруются процессы при запуске MPI-программы?
3. Что такое коммунитор? Дайте определение.
4. Как производится запуск mpi-программы с помощью команды mpirun?
5. Можно ли в качестве значения тэга в команде отправки сообщения (двухточечный обмен) передать значение номера процесса в коммуниторе? Как выполнить приём данного сообщения?
6. Какие группы функций обмена сообщениями существуют в MPI?
7. Может ли размер буфера приёма быть меньше буфера отправки?
8. Может ли начальный адрес буфера рассылки совпадать с адресом буфера процессополучателя при вызове функции MPI Scatter?

9. Можно ли обращаться к MPI - функциям после вызова MPI_Finalize?
10. Возвращает ли функция MPI_Send код ошибки, номер процесса, которому адресована передача?
11. В чем различие между блокирующим и неблокирующим обменом?
12. Можно ли в качестве тегов при послыке различных сообщений в программе использовать одно и то же число?
13. Как избежать тупиковые ситуации при использовании функций обмена?
14. В каком порядке располагает сообщения корневой процесс при выполнении операции сборки данных MPI_Gather?
15. Назовите и опишите состав и назначение параметров функций двухточечного обмена MPI.
16. В каком порядке производится операция редукции над данными из разных процессов при вызове функции MPI_Reduce?
17. Что понимается под «сообщением» в MPI? Какими атрибутами обладает сообщение?
18. Как в MPI определить номер процесса? Количество процессов?
19. Что такое MPI_COMM_WORLD?
20. Сколько процессов получают данные при использовании MPI_Allreduce?
21. Как задается число процессов при запуске с mpirun?
22. Назовите базовые типы данных в MPI (для языка C).
23. Какой коммунитор определен после выполнения функции MPI_Init?
24. Может ли значение переменной size после вызова функции MPI_Comm_Size быть равным нулю?
25. Что означает блокирующий обмен сообщениями?
26. Какими атрибутами обладает каждое посылаемое сообщение в MPI?
27. После вызова какой MPI-функции начинается параллельная часть программы?
28. Укажите синтаксические отличия MPI-программы от обычной последовательной на языке C/C++.
29. В каком случае использование функций MPI_Send и MPI_Recv приведет к дедлоку (тупиковой ситуации)?
30. Что означает асинхронный обмен сообщениями?
31. Что происходит при вызове функции MPI_BARRIER? Опишите работу процессов.
32. Какие коммуниторы всегда создаются при старте MPI - программы?
33. Как узнать число запущенных процессов приложения?
34. Как можно принять любое сообщение от любого процесса?
35. Что гарантирует блокировка при передаче/приеме сообщений?
36. Можно ли использовать функцию MPI_Recv, если не известен отправитель сообщения или тэг сообщения?
37. С помощью какой функции MPI можно определить ранг процесса?
38. Можно ли использовать функции MPI до вызова MPI_Init?
39. Назовите основные коллективные операции передачи и приема в MPI.
40. Назовите основные коллективные операции редукции в MPI.
41. Сколько процессов коммунитора участвуют в коллективных операциях?
42. Может ли MPI-программа продолжать работу после аварийного завершения одного из процессов?
43. Что такое стандарт MPI?
44. Что выполняет функция MPI_Scan?
45. Как откомпилировать MPI-программу на языке C/C++?
46. Что такое SPMD?
47. Как засечь время работы MPI-программы?
48. Что выполняет функция MPI_Sendrecv?
49. Что выполняет функция MPI_Bcast?
50. Что выполняет функция MPI_Scatter?

51. Какие операции редукции в функции MPI_Reduce известны вам?

52. Есть ли функция MPI_Reduce_scatter?

Тема №4 Параллельные вычисления определенных и кратных интегралов

1. В чем заключается преимущество обобщенной формулы средних прямоугольников перед формулой трапеций?

2. Докажите, что метод средних прямоугольников имеет больший объем вычислений по сравнению с формулой трапеций.

3. Для чего нужны адаптивные алгоритмы при вычислении интегралов? Опишите принципы их работы.

4. Как построить параллельный алгоритм расчета по квадратурной формуле? Перечислите основные этапы.

5. Чем опасен блочный способ распределения вычислительной нагрузки между процессорами при вычислении интегралов?

6. Как лучше распределять подобласти вычисления интегралов между процессорами при использовании адаптивного алгоритма?

7. Перечислите основные способы вычисления кратных интегралов. Дайте их сравнительный анализ.

8. Какие методы параллельного программирования используются при вычислении кратных интегралов? Функциональная декомпозиция или декомпозиция по данным?

Тема №5 Параллельные алгоритмы линейной алгебры (умножения векторов, матрицы на вектор и матриц)

1. Опишите как с использованием общей схемы построения параллельных алгоритмов получить параллельный алгоритм скалярного произведения векторов.

2. Перечислите возможные подходы распараллеливания матрично-векторного умножения.

3. Дайте сравнительный анализ достоинств и недостатков одномерного и двумерного укрупнения/декомпозиции при построении параллельного алгоритма умножения матрицы на вектор.

4. Опишите параллельный алгоритм матрично-векторного умножения при максимально необходимом для решения этой задачи числе процессоров.

5. Чем объяснить преимущество топологии «полный граф» по сравнению с топологией «линейка» для случая $p=n^2$?

6. Опишите как работает конвейерная схема умножения матрицы на вектор при $p=2*n$.

7. Оцените время работы параллельного алгоритма конвейерной схемы умножения матрицы на вектор при $p=2*n$.

8. В чем заключается преимущество конвейерного способа решения задачи?

9. Какие способы декомпозиции задачи матрично-векторного умножения возможны при ограниченном числе используемых процессоров? Дайте сравнительный анализ.

10. Какая топологическая структура наиболее подходит для параллельного матрично-векторного умножения при ограниченном числе используемых процессоров?

11. Как обеспечить баланс загрузки процессоров при умножении матриц в случае $n \% p \neq 0$?

12. Какие способы повышения производительности умножения матриц известны?

13. Какие способы декомпозиции задачи матричного умножения возможны?

14. Можно ли рассматривать параллельные вычисления не с точки зрения быстродействия получения результата, а как средство для решения сверхбольших задач, которые не помещаются в память обычного компьютера?

15. Какие параллельные блочные алгоритмы умножения матриц известны? Преимущества и недостатки.

Тема №6 Вопросы OpenMP

1. На вычислительные системы какого типа ориентирован стандарт ОрепМР?
2. Если в программе есть несколько вложенных циклов, и директива `#pragma omp for` относится к внешнему циклу, как будет выполняться параллельная программа? Приведите пример и опишите работу программы при запуске на 2 потоках.
3. Какой параметр необходимо указывать компилятору для компиляции ОрепМР программ?
4. Что такое «инкрементальное распараллеливание»?
5. У каких компонент ОрепМР выше приоритет: у функций или переменных окружения? Приведите пример функции, опции и переменной окружения, с помощью которых можно задать количество порождаемых потоков для ОрепМР программы.
6. Может ли поток-мастер выполнить область, ассоциированную с директивой `single`?
7. Что произойдет, если несколько потоков одновременно обратятся к чтению к общей переменной?
8. Можно ли одной директивой распределить между потоками итерации сразу нескольких вложенных циклов?
9. Какое количество потоков выполняют ОрепМР-программу при старте?
10. Может ли произойти конфликт, если несколько потоков одновременно обратятся к одной и той же локальной переменной, описанной как `private`?
11. С помощью какой директивы порождается параллельная область?
12. Что такое общая (`shared`) память и собственная (`private`) память? Опишите особенности этих видов памяти.
13. Потоки с какими номерами могут выполнить директивы `single` и `master`? Какие не могут выполнить?
14. Какие номера имеют потоки в параллельной области?
15. Как откомпилировать OpenMP-приложение?
16. Напишите ОрепМР функцию для определения номера потока.
17. Чем отличаются директивы `single` и `master`?
18. Чем отличается поток-мастер от всех остальных потоков?
19. Может ли поток с номером 1 выполнить область, ассоциированную с директивой `master`?

3. Оценочные материалы итогового контроля (промежуточной аттестации) и критерии оценивания

При оценивании знаний студентов учитываются результаты индивидуального тестирования по билетам с вопросами, приведенными выше, которое проводится в середине семестра после окончания чтения лекций. Кроме того, учитываются оценки, полученные студентом в результате защиты своих отчетов по индивидуальным заданиям по следующим темам лабораторных работ:

1. Одноточечные функции передачи сообщений в MPI.
2. Коллективные функции передачи сообщений в MPI.
3. Программирование алгоритмов суммирования в MPI.
4. Параллельное вычисление определенных интегралов в MPI.
5. Параллельное вычисление кратных интегралов в MPI.
6. Параллельное вычисление кратных интегралов и умножение матриц в OpenMP.

Индивидуальные задания выполняются во время лабораторных занятий в течение семестра.

Примеры билетов (ИОПК-7.1, ИОПК-7.2).

Билет № 1

1 Тема №1 7. Сформулируйте закон Амдала. Каким будет максимальное ускорение параллельной программы, в которой 1% арифметических операций выполняется только одним процессором?

2 Тема №2 22. Опишите алгоритм модифицированной каскадной схемы суммирования элементов последовательности (МКСС).

3 Тема №2 24. Дайте оценку ускорения и эффективности МКСС.

4 Тема №4 35. Как лучше распределять подобласти вычисления интегралов между процессорами при использовании адаптивного алгоритма?

5 Тема №1 11. Самый мощный суперкомпьютер в мире имеет производительность: 1 Deltaflops или 1 Exaflops или 1 Petaflops? Дайте правильный ответ.

6 MPI 14. В каком порядке располагает сообщения корневой процесс при выполнении операции сборки данных MPI_Gather?

7 MPI 29. После вызова какой MPI-функции начинается параллельная часть программы?

8 MPI 38. Можно ли использовать функцию MPI_Recv, если не известен отправитель сообщения или тэг сообщения?

9 MPI 56. Как засечь время работы MPI-программы?

10 MPI 57. Что выполняет функция MPI_Sendrecv?

11 MPI 58. Что выполняет функция MPI_Bcast?

12 OpenMP 10. Какое количество потоков выполняют OpenMP-программу при старте?

Билет № 2

1 Тема №1 1. В каких единицах измеряется производительность компьютера?

2 Тема №2 27. Как применяется метод циклической редукции для распараллеливания линейной рекуррентной формулы первого порядка?

3 Тема №4 36. Перечислите основные способы вычисления кратных интегралов. Дайте их сравнительный анализ.

4 Тема №1 3. Что такое тест Linpack? Какую математическую задачу он решает? Каков его результат?

5 Тема №1 7. Какова производительность самого мощного суперкомпьютера в мире?

6 MPI 7. Может ли размер буфера приема быть меньше буфера отправки?

7 MPI 9. Можно ли обращаться к MPI - функциям после вызова MPI_Finalize?

8 MPI 13. Как избежать тупиковые ситуации при использовании функций обмена? Приведите пример тупиковой ситуации.

9 MPI 36. Как можно принять любое сообщение от любого процесса?

10 MPI 38. Можно ли использовать функцию MPI_Recv, если не известен отправитель сообщения или тэг сообщения?

11 MPI 52. Может ли длина буфера получателя быть большей, чем длина принимаемого сообщения? А меньшей?

12 OpenMP 2. Если в программе есть несколько вложенных циклов, и директива #pragma omp for относится к внешнему циклу, как будет выполняться параллельная программа? Приведите пример и опишите работу программы при запуске на 2 нитях.

Примеры индивидуальных заданий (ИОПК-7.1, ИОПК-7.2, ИОПК-7.3, ИОПК-7.4)

1. Одноточечные функции передачи сообщений в MPI.

На каждом процессе определить значение переменной temp. Затем с использованием функций Send и Recv организовать передачу значения temp от каждого процесса остальным и замену значения temp наибольшим. Результат распечатать.

Вычислить скалярное произведение векторов, компоненты которых равномерно распределены между активированными процессами. Результат распечатать на «0»-процессе.

2. Коллективные функции передачи сообщений в MPI.

На 0-м процессе дан массив из $3 \cdot \text{size}$ чисел, где size — количество процессов. Используя функцию `MPI_Scatter`, переслать по 3 числа в каждый процесс (включая главный) и вывести в каждом процессе полученные числа.

На процессе с номером '0' определить массив из 8 целых чисел. С помощью функций `MPI_SCATTER` разослать по 4 процессам фрагменты этого массива. Каждый процесс печатает полученные данные. Произвести сложение всех элементов массива, распределенных по процессам, с помощью функций `MPI_REDUCE` с получением результата на процессе с номером '2'. Вычисленное значение распечатать.

3. Программирование алгоритмов суммирования в MPI.

Задание №3.1 Дана последовательность числовых значений целого типа (например, $x(i)=i$, $i=1, \dots, n$). Необходимо написать MPI-программу, реализующую вычисление частичных сумм значений заданной последовательности, применив каскадную схему суммирования. Протестировать программу для различных n .

Задание №3.2 Дана последовательность числовых значений целого типа (например, $x(i)=i$, $i=1, \dots, n$). Необходимо написать MPI-программу, реализующую вычисление суммы значений заданной последовательности, применив алгоритм сдваивания. Протестировать программу для различных n .

Задание №3.3 Дана последовательность числовых значений целого типа (например, $x(i)=i$, $i=1, \dots, n$). Необходимо написать MPI-программу, реализующую вычисление суммы значений заданной последовательности, применив модифицированную схему суммирования. Протестировать программу для различных n .

4. Параллельное вычисление определенных интегралов в MPI.

Написать MPI-программу приближенного вычисления определенного интеграла с точностью $\varepsilon=10^{-10}$, используя обобщенную формулу (левых, средних, правых прямоугольников, трапеций, Симпсона, формулу Ньютона «3/8»). Для контроля точности использовать результат расчета на MathCad. Оценить ускорение и эффективность параллельной программы.

5. Параллельное вычисление кратных интегралов в MPI.

Написать MPI-программу вычисления двойного интеграла с точностью $\varepsilon=10^{-5}$, используя метод (Монте-Карло, повторного интегрирования, ячеек). Для контроля точности использовать результат расчета на MathCad. Оценить ускорение и эффективность параллельной программы.

6. Параллельное вычисление кратных интегралов и умножение матриц в OpenMP.

Написать OpenMP-программу вычисления приближенного значения интеграла $\iint f(x,y)dydx$ с помощью метода (ячеек, повторного интегрирования, Монте-Карло). С помощью OpenMP-функций, замеряющих время выполнения программы, исследовать ускорение полученных параллельных программ на узле кластера ТГУ Cyberia на двух, трех и четырех ядрах. Исследовать возможность ускорения параллельного процесса вычисления интеграла с использованием ключей `static`, `dynamic`, `guided` директивы `for`. Результат вычислений сравнить с MathCad

Исследовать время счета, ускорение и эффективность OpenMP-программ умножения матриц для вариантов циклов $i-k-j$ и $k-i-j$. С помощью ключей `dynamic` для директивы `for` найти распределение итераций по нитям с минимальным временем

выполнения. Оценку времени работы программы проводить минимум по трем ее запускам. Обосновать полученные результаты. Исследовать ускорение и эффективность параллельной программы при $n=2048$ (число строк и столбцов матриц). Результаты умножения матриц сравнить с последовательной программой.

Каждый отчет (объем несколько страниц >1) по индивидуальным заданиям должен содержать следующие пункты:

1. Постановку задачи/задание
2. Краткое изложение подхода для ее решения
3. Программу на языке C/C++ с комментариями
4. Результаты тестирования программы, демонстрирующие правильное и полное выполнение задания
5. Краткие выводы – подтверждение приобретенных компетенций

Ответы по билетам позволяют проверить ИОПК-7.1, ИОПК-7.2. Результаты защиты отчетов по индивидуальным заданиям проверяют ИОПК-7.1, ИОПК-7.2, ИОПК-7.3, ИОПК-7.4.

Критерии оценивания:

Результаты дифференциального зачета определяются оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» выставляется, если даны правильные ответы на более чем 10 вопросов билета и большинство отчетов защищено на оценку отлично, т.е. все индивидуальные задания решены без ошибок.

Оценка «хорошо» выставляется, если правильные ответы даны на более чем 8, но менее или 10 вопросов билета и/или большинство отчетов защищено на оценку хорошо. Все индивидуальные задания решены без ошибок.

Оценка «удовлетворительно» выставляется, если правильные ответы даны на более чем 6, но менее или 8 вопросов билета и/или большинство отчетов защищено на оценку удовлетворительно. Все индивидуальные задания решены без ошибок.

Оценка «неудовлетворительно» выставляется, если правильные ответы даны на менее чем на 6 вопросов билета и/или представлены отчеты с правильными решениями не по всем темам индивидуальных заданий.

4. Оценочные материалы для проверки остаточных знаний (сформированности компетенций)

Тест (ИОПК-7.1, ИОПК-7.2, ИОПК-7.3, ИОПК-7.4)

1. В каких единицах измеряется производительность суперкомпьютера?
 - А) в Гбайт/с
 - Б) в Гц
 - В) в флопсах
 - Г) Вт
2. Что является вторым этапом разработки параллельной программы?
 - А) декомпозиция
 - Б) проектирование коммуникаций
 - В) укрупнение
 - Г) планирование вычислений
3. Какая функция передачи сообщений MPI относится к коллективным?
 - А) MPI_Send
 - Б) MPI_Reduce

В) MPI_Recv

Г) MPI_Wtime

4. Какая директива OpenMP используется для распараллеливания цикла?

а) #pragma omp parallel

б) #pragma omp barrier

в) #pragma omp sections

г) #pragma omp for

Ключи: 1 В), 2 Б), 3 Б), 4 г).

Информация о разработчиках

Старченко Александр Васильевич, д-р физ.-мат. наук, профессор кафедры вычислительной математики и компьютерного моделирования ТГУ.

Данилкин Евгений Александрович, канд. физ.-мат. наук, доцент кафедры вычислительной математики и компьютерного моделирования ТГУ.

Лаева Валентина Ивановна, ст. преподаватель кафедры вычислительной математики и компьютерного моделирования ТГУ.

Лещинский Дмитрий Викторович, ст. преподаватель кафедры вычислительной математики и компьютерного моделирования ТГУ.