

## **Спецкурс "Математическое обеспечение высокопроизводительных вычислений"**

**лектор — с.н.с. Роганов В.А.  
0,5 года; 3-6 курсы; 2015/2016 уч.г.**

В курсе рассматриваются вопросы организации высокопроизводительных вычислений для современных вычислительных платформ: суперЭВМ, Grid/Cloud систем, multi/many core систем и спецвычислителей. Особое внимание уделяется вопросам масштабируемости программ и масштабируемым параллельным алгоритмам. Изучаются специфика организации высокопроизводительного счета и основные методы повышения производительности ПО: специализация, распараллеливание и низкоуровневая оптимизация кода.

Объясняется важность эквивалентных преобразований программ и методов получения параллельных версий унаследованного ПО. Изучаются операционный недетерминизм в контексте параллельных вычислений, оптимистичные стратегии синхронизации, а также общие принципы организации отказоустойчивого счета и некоторые отказоустойчивые вычислительные алгоритмы.

В части эффективного использования аппаратуры рассматриваются организация межпроцессного сетевого взаимодействия при помощи активных сообщений, а для SIMD-архитектур - методы векторизации вычислений в применении к технологии CUDA. Как перспективные рассматриваются реконфигурируемые вычислительные системы.

С практической стороны рассматриваются конструкции современного стандарта языка C++ и его параллельные диалекты; уделяется внимание профилированию программного обеспечения, а также технологиям отладки параллельных программ.

### **Экзаменационные вопросы спецкурса**

1. Задачи, решение которых требует высокопроизводительных вычислений.
2. Специфика организации высокопроизводительных программ.
3. Высокая производительность как аспект программного кода.
4. Основные пути увеличения производительности программ.
5. Современные суперЭВМ, Grid & Cloud системы, спецвычислители.
6. Операционная и денотационная семантики программ.
7. Эквивалентные преобразования программ.
8. Графы потоков данных (DataFlow) и управления (ControlFlow).
9. Анализ алгоритмов на предмет выделения параллельных гранул.
10. Статическая и динамическая специализация программ.
11. Мемоизация (табулирование) результатов вычислений.
12. Частичные вычисления и глубокая специализация программ.
13. Практические методы трансформации кода. Использование DSL и спецкомментариев.
14. Общее определение масштабируемости программ и смежные вопросы качества ПО.
15. Масштабирование и распараллеливание программ. Особенности параллельных вычислений.
16. Увеличение производительности при параллельных вычислениях. Закон Амдала. Гиперлинейное ускорение.
17. Статическое и динамическое распараллеливание программ.
18. Высокоуровневое распараллеливание программ. T-система и ее аналоги. Транзакционная

память.

19. Балансировка загрузки вычислительных узлов.
20. Логический и физический параллелизм. Диаграмма состояний процесса.
21. Блокировка и реактивация процессов. Процессы и потоки в контексте ОС.
22. Способы обмена данными между процессами. Общая память. Сообщения. MPI.
23. Технология активных сообщений.
24. Синхронизация процессов. Взаимное исключение, назначение, требования к реализации.
25. Общая классификация параллельных компьютеров и систем. Производительность параллельных компьютеров.
26. Кластерные вычислительные системы и их особенности.
27. Многопроцессорные компьютеры с общей памятью.
28. Проблемы синхронизации. «Гонки» и взаимная блокировка процессов.
29. Методы работы с общей (разделяемой) памятью.
30. Асинхронный ввод/вывод.
31. Принципы работы планировщика и быстрое переключение контекста процессов и нитей.
32. Поддержка нитей (потоков управления) на уровне ядра операционной системы.
33. POSIX API для потоков/мьютексов и параллельные конструкции стандарта языка C++.
34. Стандарт и особенности технологии OpenMP.
35. Векторизация. Использование графических ускорителей для параллельных вычислений.
36. Динамический параллелизм в технологии CUDA.
37. Ограничения и слабые места современных компьютеров.
38. Решения, которые используются сегодня в высоко- и сверхвысокопроизводительных аппаратных комплексах.
39. Перспективные технологии создания высокопроизводительных вычислительных машин.
40. Анализ производительности программного обеспечения путем изучения машинного кода. Профилирование программного обеспечения.
41. Различные приемы, используемые компилятором для повышения производительности сгенерированного потока машинных инструкций.
42. Иерархия памяти. Эффективное использование кэш-памяти. Инstrukция prefetch.
43. Учет специфики работы виртуальной памяти. Поддержка TLB на уровне ядра ОС.
44. Методы организации взаимного исключения. Атомарные операции.
45. Низкоуровневая векторизация программного кода. Расширения AVX2 и FMA.
46. Технологии параллельного программирования CUDA и OpenCL.
47. Выбор оптимального алгоритма в зависимости от структуры вычислительной системы.
48. Обзор гибридных и реконфигурируемых вычислительных систем.
49. Вопросы обеспечения отказоустойчивости вычислений.
50. Проблемы, связанные с производительностью виртуализованной аппаратуры. Паравиртуализация.