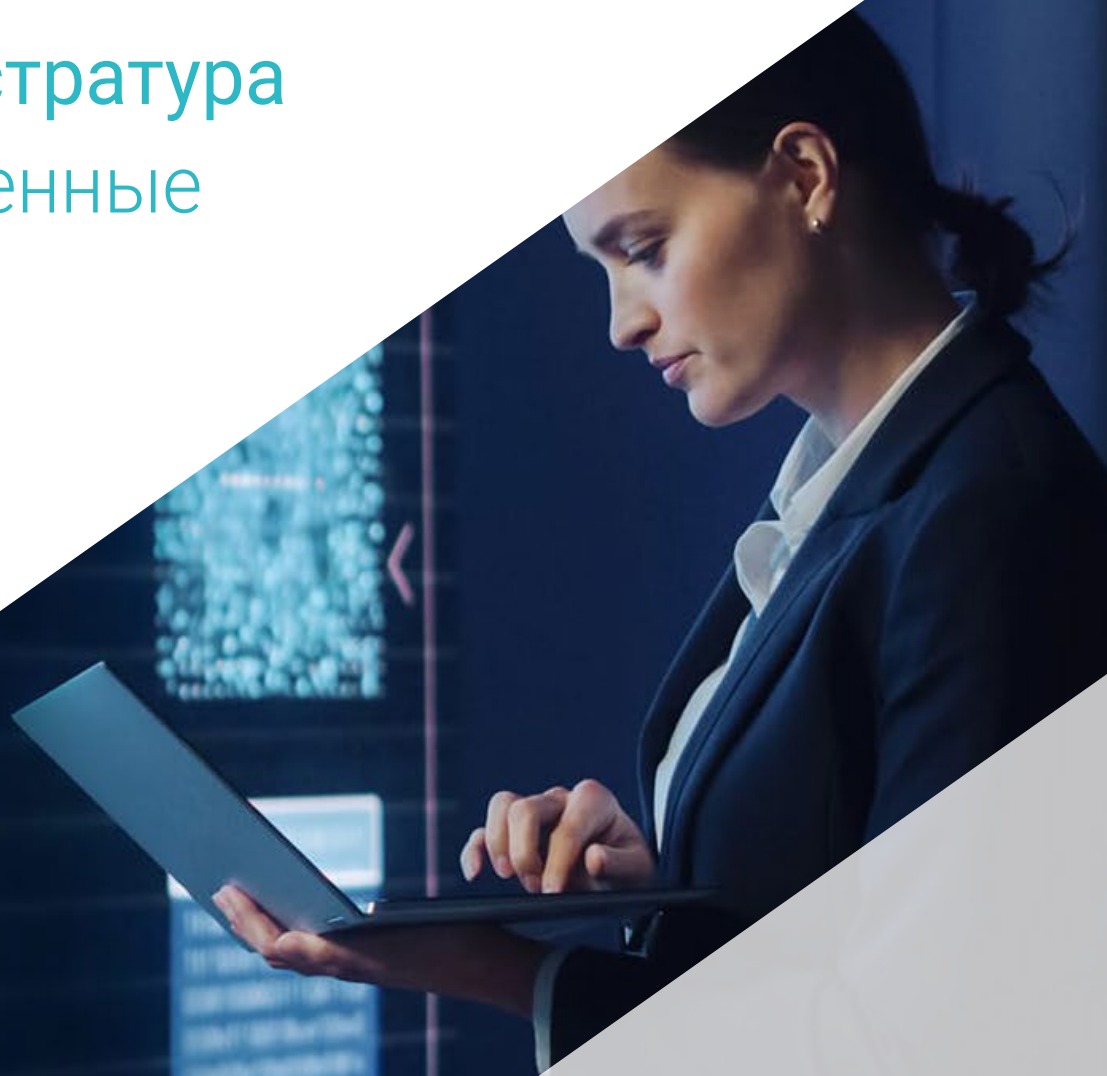


Специализированная магистратура Параллельные и распределенные вычисления





Специализированная магистратура Параллельные и распределенные вычисления

- » Формат: **онлайн**
- » Продолжительность: **12 месяцев**
- » Учебное заведение: **TECH Технологический университет**
- » Режим обучения: **16ч./неделя**
- » Расписание: **по своему усмотрению**
- » Экзамены: **онлайн**

Веб-доступ: www.techtitute.com/ru/information-technology/professional-master-degree/master-parallel-distributed-computing

Оглавление

01

Презентация

стр. 4

02

Цели

стр. 8

03

Компетенции

стр. 14

04

Руководство курса

стр. 18

05

Структура и содержание

стр. 22

06

Методология

стр. 32

07

Квалификация

стр. 40

01

Презентация

В настоящее время большинство программного обеспечения и электронных систем в той или иной форме используют параллельное или распределенное вычисление. Например, смартфоны в течение многих лет интегрировали более мощные многоядерные процессоры, а современные компьютеры уже обычно оснащены многоядерными процессорами. С другой стороны, распределенные вычисления способствовали развитию таких областей исследований, как обработка *больших данных*, которая стала неотъемлемой в таких сферах, как социальные сети, корпоративные сети и многопользовательские онлайн-игры. Все это отражает важность этих двух подходов к программированию, поэтому ТЕСН создал эту полноценную университетскую программу, в которой специалисты в области информатики смогут познакомиться с преимуществами и основными применениями параллельного и распределенного вычисления.





“

Выведите свою карьеру и резюме на новый уровень, внедрив в свою работу последние знания в области параллельного и распределенного вычисления”

Хорошее и продвинутое знание в области параллельного и распределенного вычисления может привести к профессиональному росту специалиста в области информатики, который ищет способ выделиться. Учитывая, что это сложная тема, которая может иметь множество применений, TECH обратился к команде экспертов в этой области для разработки всех материалов.

Таким образом, специалисты найдут разделы, посвященные коммуникации и координации в системах вычислений, анализу и программированию параллельных алгоритмов, распределенным системам вычислений и другим полезным вопросам. Все это изложено с современной и инновационной перспективой, основанной на собственном опыте преподавательского персонала.

Таким образом, специалисты в области информатики, успешно окончившие эту программу, имеют решающее преимущество в разработке приложений или систем в области климата, здравоохранения, *больших данных*, облачных вычислений или *блокчейна*. Кроме того, благодаря продвинутой программе, можно рассматривать возможность исследовательской деятельности в области вычислений или других связанных областей.

Кроме того, данный курс предлагается полностью в онлайн-формате, что устраняет необходимость посещения занятий или ограничений фиксированного расписания. Специалисты в области информатики могут свободно распределить учебную нагрузку в соответствии со своими интересами, совмещая изучение данной Специализированной магистратуры с другими личными или профессиональными обязанностями.

Данная **Специализированная магистратура в области параллельные и распределенные вычисления** содержит самую полную и современную образовательную программу на рынке. Основными особенностями обучения являются:

- ◆ Разработка практических кейсов, представленных экспертами в области параллельных и распределенных вычислений
- ◆ Наглядное, схематичное и исключительно практическое содержание курса предоставляет практическую информацию по тем дисциплинам, которые необходимы для осуществления профессиональной деятельности
- ◆ Практические упражнения для самопроверки, контроля и улучшения успеваемости
- ◆ Особое внимание уделяется инновационным методологиям
- ◆ Теоретические занятия, вопросы эксперту и самостоятельные работы
- ◆ Учебные материалы курса доступны с любого стационарного или мобильного устройства с выходом в интернет



Запишитесь сейчас и начните уже сейчас изучать самые последние новинки в области параллельных вычислений в облачных средах и программирования, ориентированного на распределенные вычисления”

“

Всегда будете получать консультацию от преподавательского состава, в котором работают профессионалы с большим опытом в области параллельных и распределенных вычислений”

В преподавательский состав входят профессионалы отрасли, которые вносят свой опыт работы в эту программу, а также признанные специалисты, принадлежащие к ведущим научным сообществам и престижным университетам.

Мультимедийное содержание программы, разработанное с использованием новейших образовательных технологий, позволит специалисту проходить обучение с учетом контекста и ситуации, т. е. в симулированной среде, обеспечивающей иммерсивный учебный процесс, запрограммированный на обучение в реальных ситуациях.

Структура этой программы основана на проблемно-ориентированном обучении, с помощью которого специалист должен попытаться решить различные ситуации из профессиональной практики, возникающие в течение учебного курса. В этом специалисту поможет инновационная интерактивная видеосистема, созданная признанными экспертами.

Вы получите полную поддержку от крупнейшего онлайн-образовательного учреждения в мире, TECH, с передовыми образовательными технологиями.

Не упустите возможность выделиться и продемонстрировать свой интерес к информатике настоящего и будущего.



02

Цели

В связи с тем, что информатика развивается стремительно и специалисты в этой области должны непрерывно обновлять свои знания, TECH сосредоточился на самых новых достижениях в области параллельных и распределенных вычислений. Таким образом, студенты не только осваивают передовые навыки, но также изучают множество применений таких технологий, как *блокчейн* и облачные вычисления, которые имеют важное значение в современном мире.



“

Ваша цель для профессионального роста будет намного ближе благодаря советам и ключевым моментам в области вычислений, которые вы узнаете в этой программе”



Общие цели

- ♦ Анализировать взаимодействие между различными компонентами параллельных и распределенных вычислений
- ♦ Измерять и сравнивать их производительность для анализа эффективности используемого набора компонентов
- ♦ Подробно исследовать многоплатформенную параллельную вычислительную среду для использования параллелизма на уровне задач между различными аппаратными ускорителями
- ♦ Подробно изучать текущие программные и аппаратные архитектуры
- ♦ Глубоко разрабатывать ключевые аспекты параллельных и распределенных вычислений
- ♦ Специализировать студентов в использовании параллельных и распределенных вычислений в различных областях применения



Вы ознакомитесь со всеми наиболее важными аспектами параллельных и распределенных вычислений, начиная от их особенностей в параллельности и заканчивая их многочисленными применениями”



Конкретные цели

Модуль 1. Параллелизм в параллельных и распределенных вычислениях

- ♦ Анализировать компоненты обработки: процессор или память
- ♦ Углубить знания в области архитектуры параллелизма
- ♦ Анализировать различные формы параллелизма с точки зрения процессора

Модуль 2. Параллельная декомпозиция в параллельных и распределенных вычислениях

- ♦ Анализировать важность декомпозиции процессов на параллельные процессы при решении вычислительных задач
- ♦ Изучать различные примеры для демонстрации применения и использования параллельных вычислений и их декомпозиции
- ♦ Излагать процедуры и инструменты, позволяющие выполнять процессы параллельно с целью достижения наилучшей производительности
- ♦ Получать специализированные знания для определения сценариев декомпозиции процессов на параллельные процессы и выбора их применения

Модуль 3. Коммуникация и координация в системах вычислений

- ♦ Анализировать различные архитектуры и модели распределенных систем
- ♦ Определять характеристики параллельных и распределенных систем
- ♦ Углубиться в изучение различных видов коммуникаций, которые происходят на уровне процессов
- ♦ Изучать удаленные коммуникации, направленные на потоки, сообщения и многократную передачу сообщений вместе с примерами и более актуальными соображениями
- ♦ Устанавливать виды коммуникаций, которые становятся все более популярными, их преимущества и ограничения
- ♦ Разрабатывать процессы выбора алгоритмов, применяемых для обслуживания имен, синхронизации часов, координации и согласования между элементами системы
- ♦ Собирать сценарии, в которых используются различные типы коммуникационных технологий, улучшающих производительность и масштабируемость

Модуль 4. Анализ и программирование параллельных алгоритмов

- ♦ Исследовать различные парадигмы параллельного программирования
- ♦ Изучать самые современные инструменты для реализации параллельного программирования
- ♦ Анализировать параллельные алгоритмы для фундаментальных проблем
- ♦ Конкретизировать проектирование и анализ параллельных алгоритмов
- ♦ Разрабатывать параллельные алгоритмы и реализовывать их с использованием MPI, OpenMP, OpenCL/CUDA

Модуль 5. Параллельные архитектуры

- ♦ Анализировать основные архитектуры компьютеров
- ♦ Углубить знания в области ключевых аспектов, таких как процесс, сервис и поток выполнения
- ♦ Управлять процессами в операционной системе
- ♦ Использовать классы для запуска и управления процессами

Модуль 6. Параллельная производительность

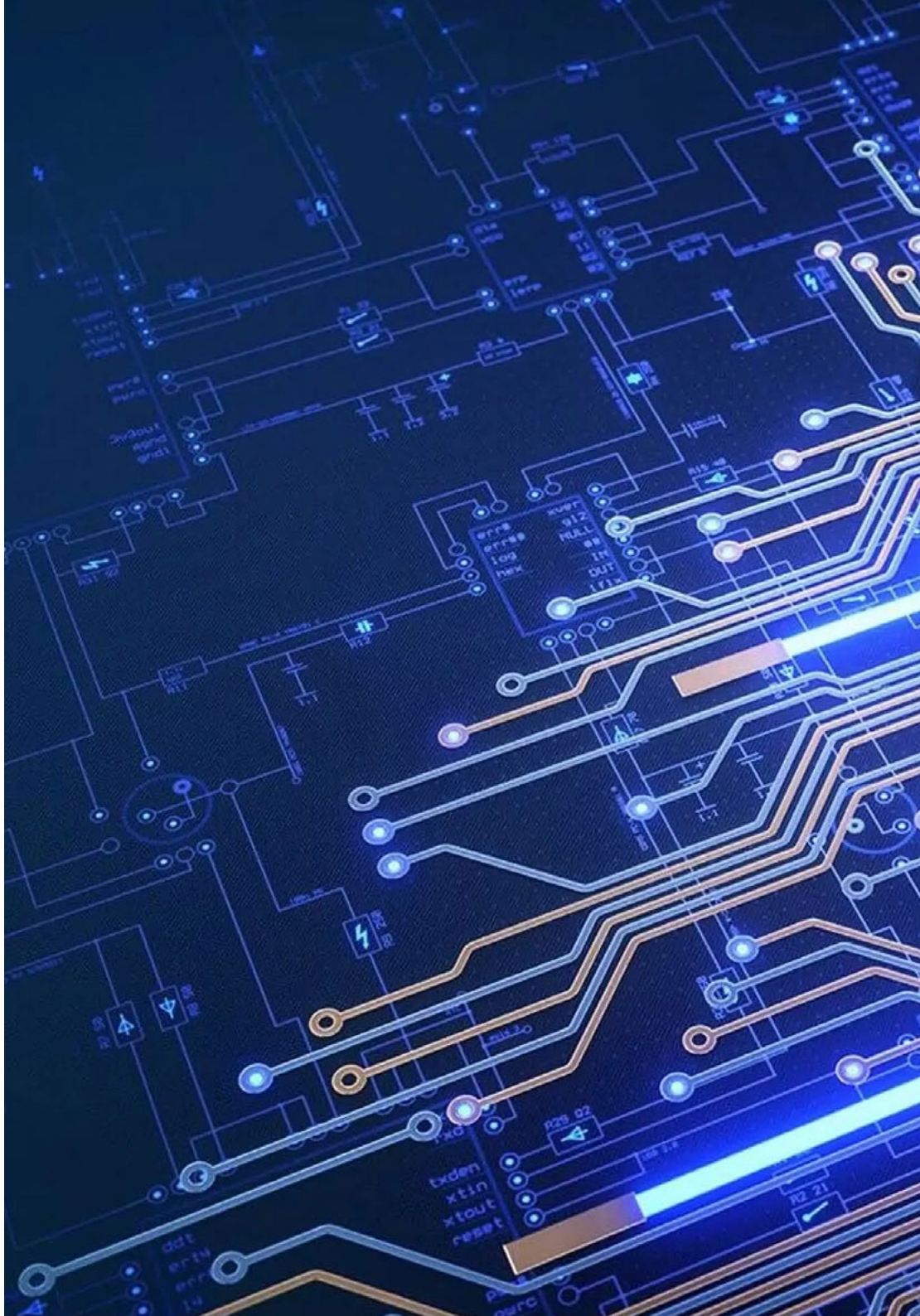
- ♦ Анализировать аспекты параллельных алгоритмов, которые влияют на их производительность и масштабируемость
- ♦ Определять основные показатели производительности и масштабируемости параллельных алгоритмов
- ♦ Изучать основные методы сравнения параллельных алгоритмов
- ♦ Определять ограничения, которые аппаратное обеспечение накладывает на параллелизацию
- ♦ Определять лучшие практики для производительности параллельных программ с общей памятью, для производительности параллельных программ с передачей сообщений, для производительности гибридных параллельных программ и для производительности параллельных программ с гетерогенным вычислением
- ♦ Собирать наиболее продвинутые инструменты для анализа производительности параллельных алгоритмов
- ♦ Представлять основные шаблоны параллельной обработки
- ♦ Разрабатывать надежную процедуру для определения высокопроизводительных параллельных программ

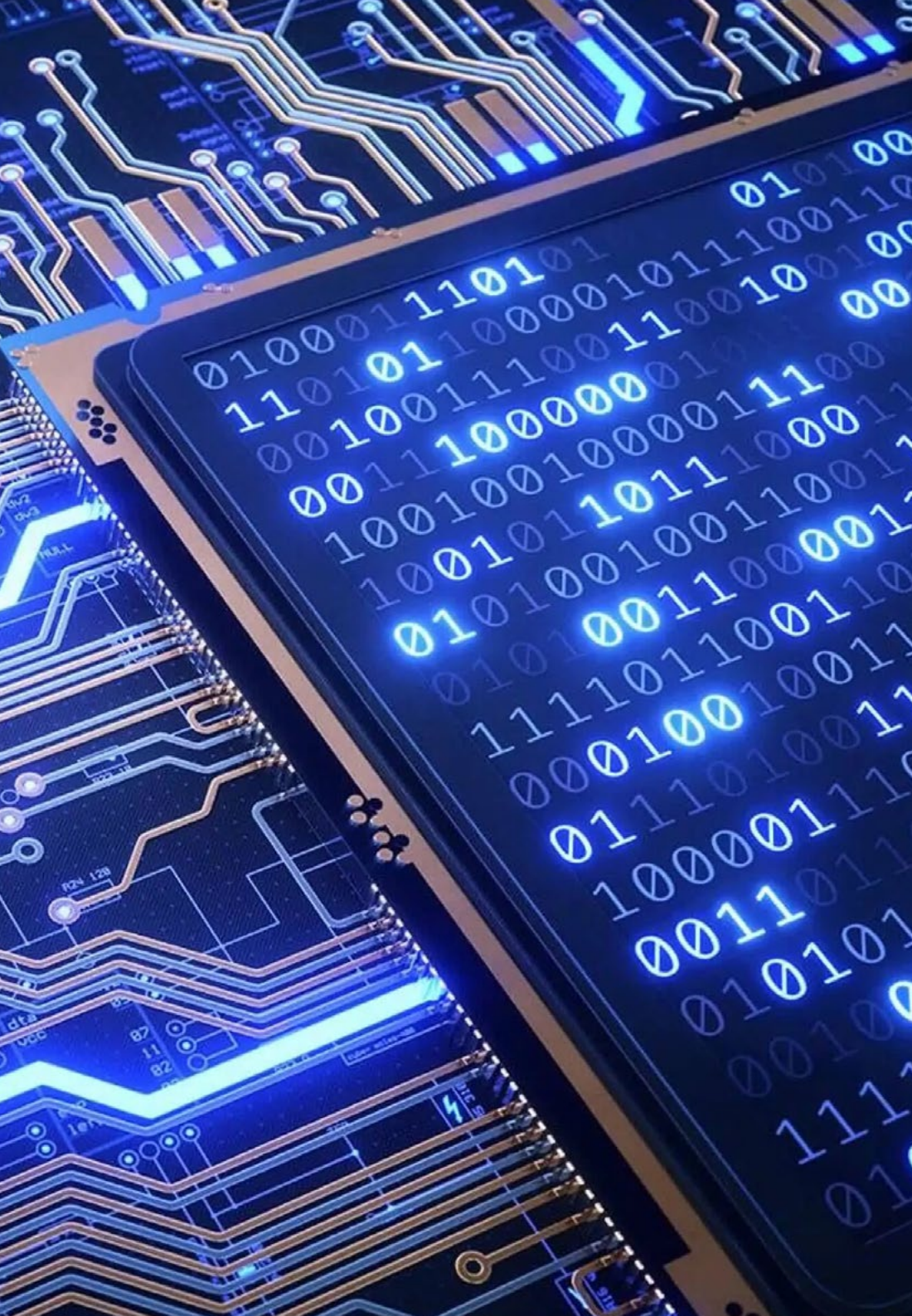
Модуль 7. Распределенные вычислительные системы

- ◆ Разрабатывать ключевые элементы распределенной системы
- ◆ Изучать элементы безопасности, применяемые в распределенных системах и их необходимость
- ◆ Представлять различные типы распределенных систем, которые чаще всего используются, их особенности, функциональность и проблемы, которые необходимо решить
- ◆ Демонстрировать теорему CAP, применимую к распределенным системам: *Consistency* (согласованность), *Availability* (доступность) и *Partition Tolerance* (устойчивость к сбоям)

Модуль 8. Параллельные вычисления в облачных средах

- ◆ Разрабатывать парадигму облачных вычислений
- ◆ Определять различные подходы на основе уровня автоматизации и услуги
- ◆ Анализировать основные компоненты архитектуры облачных вычислений
- ◆ Определять различия с архитектурой *On-Premise*
- ◆ Исследовать различные варианты развертывания *облачных* вычислений: *Multi-Cloud*, *Hybrid Cloud*
- ◆ Углубить знания в области преимуществ, присущих облачным вычислениям
- ◆ Разрабатывать принципы экономики облачных вычислений: переход от CAPEX к OPEX
- ◆ Оценивать коммерческие предложения различных *облачных* провайдеров
- ◆ Оценивать возможности суперкомпьютерных вычислений в облаке
- ◆ Изучать вопросы безопасности в облачных вычислениях





Модуль 9. Модели и формальная семантика. Программирование, ориентированное на распределенные вычисления

- ◆ Идентифицировать преимущества формальной семантики
- ◆ Исследовать, как формальная семантика помогает в программировании, ориентированном на распределенные вычисления
- ◆ Сосредоточиться на возможностях применения формальной семантики в программировании, ориентированном на распределенные вычисления
- ◆ Разрабатывать основные инструменты для обеспечения жизнеспособности проектов, использующих эту технологию
- ◆ Идентифицировать языки программирования в контексте семантической модели
- ◆ Определять, как эти семантические модели помогают с языками программирования
- ◆ Оценивать и сравнивать модели вычислений
- ◆ Сосредоточиться на использовании распределенных моделей
- ◆ Представлять наиболее передовые инструменты рынка для проектов

Модуль 10. Применение параллельных и распределенных вычислений

- ◆ Продемонстрировать значительный вклад применения параллельных и распределенных вычислений в нашей среде
- ◆ Определять основные архитектуры, используемые на рынке
- ◆ Оценивать выгоды этих примеров использования
- ◆ Представлять успешные решения на рынке
- ◆ Демонстрировать, почему это важно для оценки изменения климата
- ◆ Определять текущую важность графических процессоров (GPU)
- ◆ Показывать влияние этой технологии на электрические сети
- ◆ Исследовать распределенные двигатели для обслуживания наших клиентов
- ◆ Изучать преимущества распределенных двигателей для достижения выгоды в наших компаниях
- ◆ Представлять примеры баз данных в памяти и их важность
- ◆ Исследовать, как эти модели помогают в медицине

03

Компетенции

С должными знаниями в этой области информатики профессионалы могут выделяться и возглавлять целые программные команды. Примером этому являются преподаватели, которые имеют значительный опыт работы на руководящих должностях и руководстве множеством различных проектов в области информатики. Именно поэтому Специализированная магистратура в области параллельных и распределенных вычислений не только фокусируется на ключевых аспектах этой области, но и развивает специалиста в области информатики в ряде уникальных и выдающихся компетенций, которые помогут выделяться в ежедневной работе.



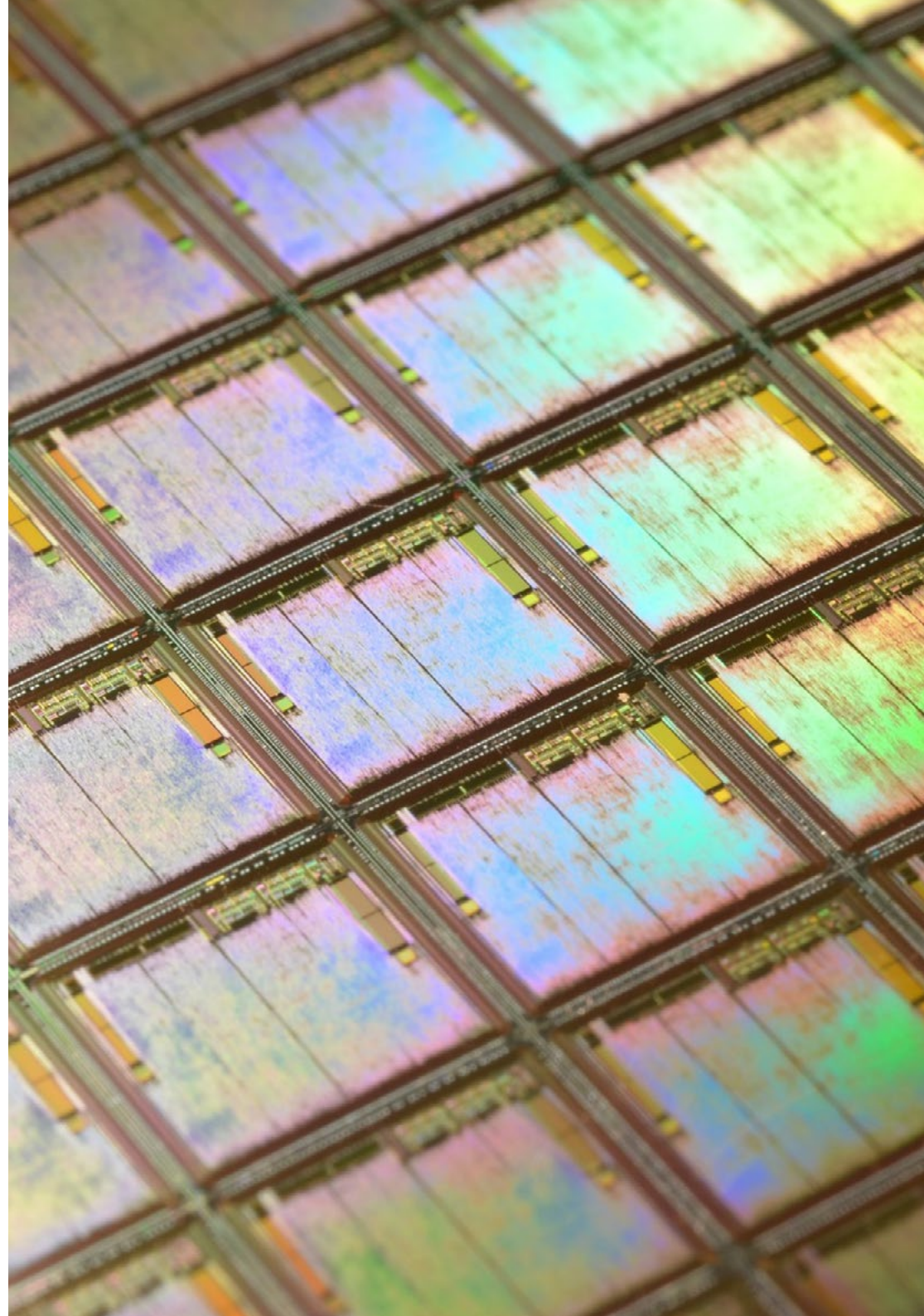
“

Данная программа станет ключом к вашему стремительному продвижению к руководящим должностям в программировании и разработке программного обеспечения любого типа”



Общие профессиональные навыки

- ◆ Развивать специализированные знания о различных уровнях параллелизма
- ◆ Анализировать стратегию параллелизации на основе метрик производительности
- ◆ Определять основные характеристики параллельных и распределенных вычислений перед изучением коммуникации и координации, которая происходит между их компонентами
- ◆ Доказывать, что в таких системах могут происходить коммуникация между процессами, удаленные вызовы или косвенные коммуникации
- ◆ Определять аспекты, негативно влияющие на производительность параллельных приложений
- ◆ Анализировать передовые техники оптимизации параллельного кода, оптимизации коммуникаций в системах с распределенной памятью, управления аффинностью, балансировки нагрузки и управления параллельным вводом/выводом
- ◆ Изучать гибридные модели программирования для систем с несколькими аппаратными ускорителями и модели программирования для систем с общей/распределенной памятью





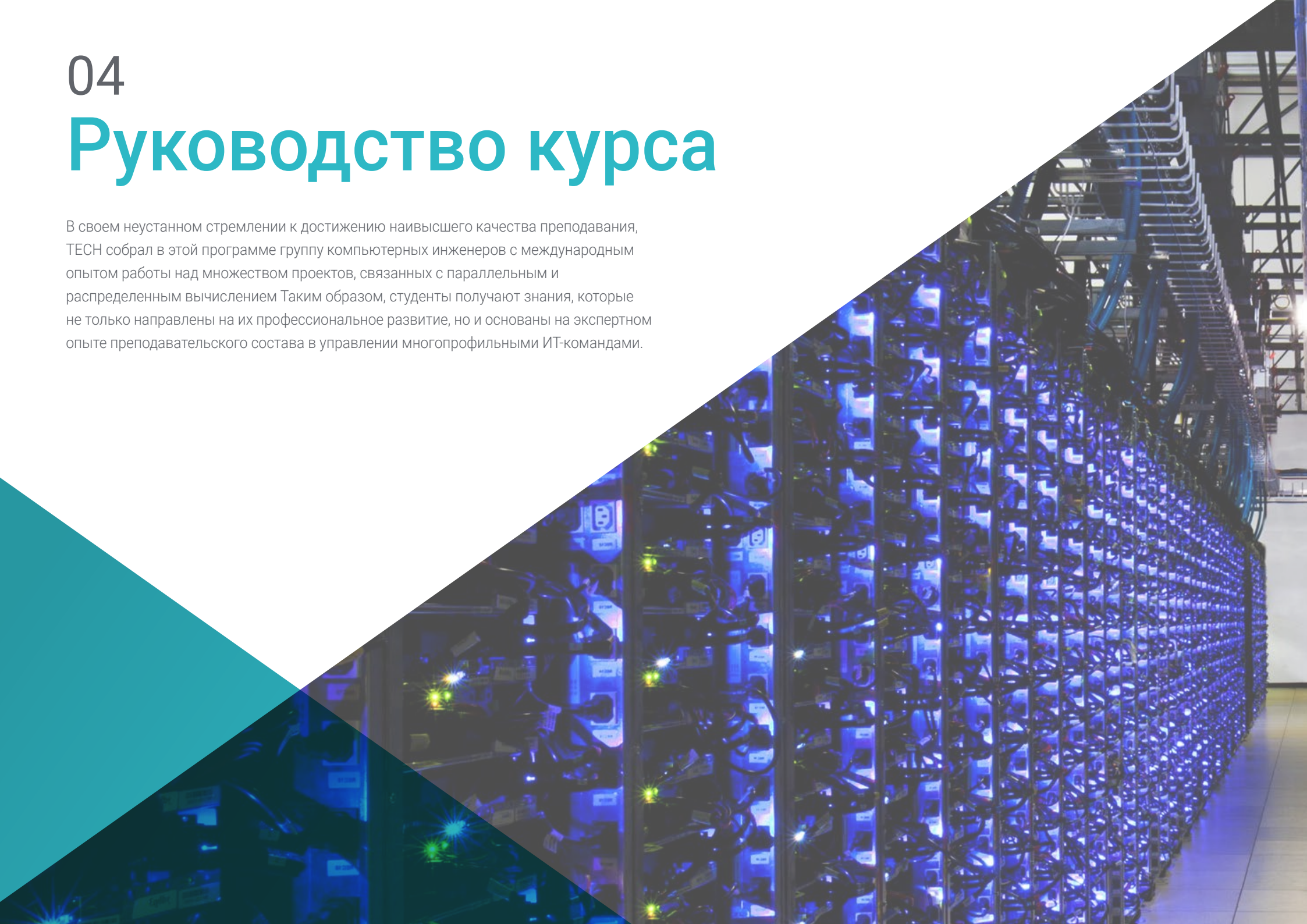
Профессиональные навыки

- ◆ Определять проблемы масштабируемости и производительности, которые могут быть решены путем параллельной декомпозиции процессов
- ◆ Анализировать особенности предложений по параллельному использованию общей памяти, передаче сообщений и параллельному использованию графических процессоров, а также гибриднему сценарию
- ◆ Установить необходимость устойчивости процессов и модели акторов для решения современных проблем вычислений
- ◆ Приводить примеры или случаи, где параллельная декомпозиция была успешно применена для повышения производительности или масштабируемости
- ◆ Анализировать и проектировать параллельные алгоритмы
- ◆ Компилировать основные примитивы MPI, OpenMP, OpenCL/CUDA
- ◆ Исследовать векторные и матричные процессы
- ◆ Анализировать параллельное и распределенное программирование, языки программирования, инструменты и шаблоны проектирования
- ◆ Определять элементы, обеспечивающие взаимодействие распределенных сетей
- ◆ Обосновывать шаги проектирования распределенной системы
- ◆ Оценивать различные типы репликации данных в существующих системах
- ◆ Собирать практические подходы к обеспечению безопасности

04

Руководство курса

В своем неустанном стремлении к достижению наивысшего качества преподавания, ТЕСН собрал в этой программе группу компьютерных инженеров с международным опытом работы над множеством проектов, связанных с параллельным и распределенным вычислением. Таким образом, студенты получают знания, которые не только направлены на их профессиональное развитие, но и основаны на экспертном опыте преподавательского состава в управлении многопрофильными ИТ-командами.



“

Развивайте свой учебный и профессиональный потенциал с лучшими преподавателями, которые будут поддерживать вас на протяжении всего обучения”

Руководство



Г-н Олайа Бональ, Мартин

- ♦ Старший менеджер в области блокчейн-практики в компании EY
- ♦ Технический специалист по блокчейн-клиентам в IBM
- ♦ Директор по архитектуре в компании Blocknitive
- ♦ Координатор группы по нереляционным распределенным базам данных в wedoIT (дочерняя компания IBM)
- ♦ Архитектор инфраструктуры в Bankia
- ♦ Руководитель отдела верстки в компании T-Systems
- ♦ Координатор отдела в Bing Data España S.L.

Преподаватели

Г-н Вильот Гисан, Пабло

- ♦ Директор по информационным технологиям, технический специалист и основатель компании New Tech & Talent
- ♦ Эксперт в области технологий в KPMG, Испания
- ♦ Архитектор *блокчейна* в Everis
- ♦ Разработчик J2EE, отдел коммерческой логистики в Inditex
- ♦ Степень бакалавра в области компьютерной инженерии в Университете Ла-Коруньи
- ♦ Сертификат от Microsoft по программе MSCA: *Облачная платформа*

Г-н Госало Фернандес, Хуан Луис

- ♦ Менеджер по продуктам на основе блокчейна для Open Canarias
- ♦ Директор по блокчейн DevOps в Аластрии
- ♦ Директор по технологиям уровня обслуживания в Santander, Испания
- ♦ Руководитель направления разработки мобильных приложений Tinkerlink в компании Cronos Telecom
- ♦ Директор по технологиям управления ИТ-услугами в Barclays Bank, Испания
- ♦ Степень бакалавра в области высшей компьютерной инженерии в UNED
- ♦ Специализация в области *глубокого обучения* в DeepLearning.ai

Д-р Бланко, Эдуардо

- ◆ Специалист в области компьютерных наук
- ◆ Преподаватель в Университете Симона Боливара
- ◆ Степень доктора в области вычислений в Университете Симона Боливара
- ◆ Компьютерный инженер в Университете Симона Боливара
- ◆ Степень магистра в области вычислений в Университете Симона Боливара

Г-н Гомес Гомес, Борха

- ◆ Руководитель отдела развития бизнеса по облачным инновациям в Oracle
- ◆ Руководитель отдела блокчейна и предпродажных архитектурных решений в Paradigma Digital
- ◆ Архитектор и старший ИТ-консультант в компании Atmira
- ◆ Архитектор SOA и консультант в TCP SI
- ◆ Аналитик и консультант в Everis
- ◆ Степень бакалавра в области компьютерной инженерии в Мадридском университете Комплутенсе
- ◆ Степень магистра в области компьютерной инженерии в Мадридском университете Комплутенсе

Д-р Карратала Саес, Росио

- ◆ Исследователь, специализирующийся в области компьютерных наук
- ◆ Преподаватель университетских курсов по информатике
- ◆ Степень доктора в области компьютерных наук в Университете Хауме I
- ◆ Степень бакалавра в области вычислительной математики в Университете Хауме I
- ◆ Степень магистра в области параллельных и распределенных вычислений в Политехническом университете Валенсии
- ◆ Специализированные курсы, связанные с компьютерными науками, математикой и инструментами для академических исследований

Д-р Альмендрас Арусаме, Луис Фернандо

- ◆ Инженер по данным и бизнес-аналитике Solutio Group, Мадрид
- ◆ Инженер по данным в компании Indizen
- ◆ Инженер по данным и *бизнес-аналитике* в компании Tecnologia y Personas
- ◆ Инженер по поддержке баз данных, *больших данных* и *бизнес-аналитики* в компании Equinix
- ◆ Инженер по данным. Jalasoft
- ◆ Менеджер по продуктам и ответственный за отдел бизнес-аналитики в компании Goja
- ◆ Заместитель менеджера по бизнес-аналитике VIVA Nuevatel PC's
- ◆ Ответственный за хранилище данных и отдела больших данных в компании Viva
- ◆ Руководитель разработки программного обеспечения в компании Intersoft
- ◆ Степень бакалавра в области информатики в Университете Майор-де-Сан-Симон
- ◆ Степень доктора в области компьютерной инженерии Мадридский университет Комплутенсе
- ◆ Степень магистра в области компьютерной инженерии в Мадридском университете Комплутенсе
- ◆ Степень магистра в области информационных систем и управления технологиями, Университет Майор-де-Сан-Симон
- ◆ Международный инструктор: Oracle Database. Proydesa — Oracle, Аргентина
- ◆ Сертификат профессионала в области управления проектами Информационно-просветительский консалтинг, Чили

05

Структура и содержание

Чтобы облегчить процесс обучения и приобретения компетенций, TECH включил в данную университетскую программу наиболее эффективную педагогическую методологию. Благодаря методологии *Relearning*, студенты значительно сокращают время, затрачиваемое на освоение основных знаний программы, а также укрепляют их благодаря большому количеству аудиовизуального материала, дополнительной литературы и практических упражнений.



“

Практические задания, основанные на реальных случаях, и детальные видеоматериалы, созданные преподавателями, станут ключом к вашему успеху в данной университетской программе”

Модуль 1. Параллелизм в параллельных и распределенных вычислениях

- 1.1. Параллельная обработка
 - 1.1.1. Параллельная обработка
 - 1.1.2. Параллельная обработка в вычислительной технике. Цель
 - 1.1.3. Параллельная обработка. Анализ
- 1.2. Параллельная система
 - 1.2.1. Параллельная система
 - 1.2.2. Уровни параллелизма
 - 1.2.3. Состав параллельной системы
- 1.3. Архитектуры процессоров
 - 1.3.1. Сложность процесса
 - 1.3.2. Архитектура процессора. Режим работы
 - 1.3.3. Архитектура процессора. Организация памяти
- 1.4. Сети в параллельной обработке
 - 1.4.1. Режим работы
 - 1.4.2. Стратегия управления
 - 1.4.3. Методы коммутации
 - 1.4.4. Топология
- 1.5. Параллельные архитектуры
 - 1.5.1. Алгоритмы
 - 1.5.2. Соединение
 - 1.5.3. Коммуникация
- 1.6. Производительность параллельных вычислений
 - 1.6.1. Эволюция производительности
 - 1.6.2. Показатели *производительности*
 - 1.6.3. Параллельные вычисления. Тематические исследования
- 1.7. Таксономия Флинна
 - 1.7.1. MIMD: общая память
 - 1.7.2. MIMD: распределенная память
 - 1.7.3. MIMD: гибридные системы
 - 1.7.4. Поток данных

- 1.8. Формы параллелизма: TLP (*Thread Level Paralelism*)
 - 1.8.1. Формы параллелизма: TLP (*Thread Level Paralelism*)
 - 1.8.2. *Крупное зерно*
 - 1.8.3. *Мелкое зерно*
 - 1.8.4. SMT
- 1.9. Формы параллелизма: DLP (*Data Level Paralelism*)
 - 1.9.1. Формы параллелизма: DLP (*Data Level Paralelism*)
 - 1.9.2. *Обработка коротких векторов*
 - 1.9.3. *Векторные процессоры*
- 1.10. Формы параллелизма: ILP (*Instruction Level Paralelism*)
 - 1.10.1. Формы параллелизма: ILP (*Instruction Level Paralelism*)
 - 1.10.2. Сегментированный процессор
 - 1.10.3. Суперскалярный процессор
 - 1.10.4. Процессор *Very Long Instruction Word* (VLIW)

Модуль 2. Параллельная декомпозиция в параллельных и распределенных вычислениях

- 2.1. Параллельная декомпозиция
 - 2.1.1. Параллельная обработка
 - 2.1.2. Архитектуры
 - 2.1.3. Суперкомпьютеры
- 2.2. Параллельное оборудование и параллельное программное обеспечение
 - 2.2.1. Последовательные системы
 - 2.2.2. Параллельное оборудование
 - 2.2.3. Параллельное программное обеспечение
 - 2.2.4. Ввод и вывод
 - 2.2.5. Производительность
- 2.3. Параллельная масштабируемость и повторяющиеся проблемы производительности
 - 2.3.1. Параллелизм
 - 2.3.2. Параллельная масштабируемость
 - 2.3.3. Повторяющиеся проблемы производительности

- 2.4. Параллелизм общей памяти
 - 2.4.1. Параллелизм общей памяти
 - 2.4.2. OpenMP и Pthreads
 - 2.4.3. Параллелизм общей памяти. Примеры
- 2.5. Графический процессор (GPU)
 - 2.5.1. Графический процессор (GPU)
 - 2.5.2. Архитектура вычислительных унифицированных устройств (CUDA)
 - 2.5.3. Архитектура вычислительных унифицированных устройств. Примеры
- 2.6. Системы передачи сообщений
 - 2.6.1. Системы передачи сообщений
 - 2.6.2. MPI. Интерфейс передачи сообщений
 - 2.6.3. Системы передачи сообщений. Примеры
- 2.7. Гибридное распараллеливание с MPI и OpenMP
 - 2.7.1. Гибридное программирование
 - 2.7.2. Модели программирования MPI/OpenMP
 - 2.7.3. Гибридная декомпозиция и отображение
- 2.8. Вычисления MapReduce
 - 2.8.1. Hadoop
 - 2.8.2. Другие вычислительные системы
 - 2.8.3. Параллельные вычисления. Примеры
- 2.9. Акторная модель и реактивные процессы
 - 2.9.1. Акторная модель
 - 2.9.2. Реактивные процессы
 - 2.9.3. Акторы и реактивные процессы. Примеры
- 2.10. Сценарии параллельных вычислений
 - 2.10.1. Обработка звука и изображений
 - 2.10.2. Статистика/добыча данных
 - 2.10.3. Параллельная сортировка
 - 2.10.4. Параллельные матричные операции

Модуль 3. Коммуникация и координация в системах вычислений

- 3.1. Параллельные и распределенные вычислительные процессы
 - 3.1.1. Параллельные и распределенные вычислительные процессы
 - 3.1.2. Процессы и потоки
 - 3.1.3. Виртуализация
 - 3.1.4. Клиенты и серверы
- 3.2. Связь параллельных вычислений
 - 3.2.1. Параллельные вычисления
 - 3.2.2. Многоуровневые протоколы
 - 3.2.3. Связь параллельных вычислений. Типология
- 3.3. Удаленный вызов процедур
 - 3.3.1. Как работает RPC (*удаленный вызов процедур*)
 - 3.3.2. Передача параметров
 - 3.3.3. Асинхронный RPC
 - 3.3.4. Удаленная процедура. Примеры
- 3.4. Коммуникация, ориентированная на сообщения
 - 3.4.1. Переходная коммуникация, ориентированная на сообщения
 - 3.4.2. Постоянная коммуникация, ориентированная на сообщения
 - 3.4.3. Коммуникация, ориентированная на сообщения. Примеры
- 3.5. Коммуникация, ориентированная на поток
 - 3.5.1. Поддержка непрерывных носителей
 - 3.5.2. Поток и качество обслуживания
 - 3.5.3. Синхронизация потока
 - 3.5.4. Коммуникация, ориентированная на поток. Примеры
- 3.6. Многоадресная коммуникация
 - 3.6.1. Многоадресная передача на уровне приложения
 - 3.6.2. Распространение данных на основе слухов
 - 3.6.3. Многоадресная коммуникация. Примеры

- 3.7. Другие виды коммуникации
 - 3.7.1. Удаленный вызов метода
 - 3.7.2. Веб-сервисы / SOA / REST
 - 3.7.3. Оповещение о событиях
 - 3.7.4. Мобильные агенты
 - 3.8. Служба имен
 - 3.8.1. Службы имен компьютеров
 - 3.8.2. Службы имен и система доменных имен
 - 3.8.3. Службы каталогов
 - 3.9. Синхронизация
 - 3.9.1. Синхронизация часов
 - 3.9.2. Логические часы, взаимное исключение и глобальное позиционирование узлов
 - 3.9.3. Выбор алгоритмов
 - 3.10. Коммуникация. Координация и соглашение
 - 3.10.1. Координация и соглашение
 - 3.10.2. Координация и соглашение. Консенсус и проблемы
 - 3.10.3. Коммуникация и координация. Современность
- Модуль 4. Анализ и программирование параллельных алгоритмов**
- 4.1. Параллельные алгоритмы
 - 4.1.1. Декомпозиция задачи
 - 4.1.2. Зависимости данных
 - 4.1.3. Неявный и явный параллелизм
 - 4.2. Парадигмы параллельного программирования
 - 4.2.1. Параллельное программирование с общей памятью
 - 4.2.2. Параллельное программирование с распределенной памятью
 - 4.2.3. Гибридное параллельное программирование
 - 4.2.4. Гетерогенные вычисления — CPU + GPU
 - 4.2.5. Квантовые вычисления. Новые модели программирования с неявным параллелизмом
 - 4.3. Параллельное программирование с общей памятью
 - 4.3.1. Модели параллельного программирования с общей памятью
 - 4.3.2. Параллельные алгоритмы с общей памятью
 - 4.3.3. Библиотеки параллельного программирования с общей памятью
 - 4.4. OpenMP
 - 4.4.1. OpenMP
 - 4.4.2. Выполнение и отладка программ с использованием OpenMP
 - 4.4.3. Параллельные алгоритмы с общей памятью в OpenMP
 - 4.5. Параллельное программирование с передачей сообщений
 - 4.5.1. Примитивы передачи сообщений
 - 4.5.2. Коммуникации и коллективные вычислительные операции
 - 4.5.3. Параллельные алгоритмы передачи сообщений
 - 4.5.4. Библиотеки для параллельного программирования с передачей сообщений
 - 4.6. *Интерфейс передачи сообщений (MPI)*
 - 4.6.1. *Интерфейс передачи сообщений (MPI)*
 - 4.6.2. Выполнение и отладка MPI-программ
 - 4.6.3. Параллельные алгоритмы передачи сообщений с MPI
 - 4.7. Гибридное параллельное программирование
 - 4.7.1. Гибридное параллельное программирование
 - 4.7.2. Выполнение и отладка гибридно-параллельных программ
 - 4.7.3. Гибридные параллельные алгоритмы MPI-OpenMP
 - 4.8. Параллельное программирование с использованием гетерогенных вычислений
 - 4.8.1. Параллельное программирование с использованием гетерогенных вычислений
 - 4.8.2. CPU vs. GPU
 - 4.8.3. Параллельные алгоритмы на гетерогенных вычислениях
 - 4.9. OpenCL и CUDA
 - 4.9.1. OpenCL vs. CUDA
 - 4.9.2. Выполнение и отладка параллельных программ с использованием гетерогенных вычислений
 - 4.9.3. Параллельные алгоритмы на гетерогенных вычислениях
 - 4.10. Проектирование параллельных алгоритмов
 - 4.10.1. Проектирование параллельных алгоритмов
 - 4.10.2. Проблема и контекст
 - 4.10.3. Автоматическое распараллеливание vs. Ручное распараллеливание
 - 4.10.4. Разбиение проблемы на разделы
 - 4.10.5. Компьютерные коммуникации

Модуль 5. Параллельные архитектуры

- 5.1. Параллельные архитектуры
 - 5.1.1. Параллельные архитектуры. Классификация
 - 5.1.2. Источники параллелизма
 - 5.1.3. Параллелизм и процессоры
- 5.2. Производительность параллельных систем
 - 5.2.1. Метрики и показатели производительности
 - 5.2.2. *Ускорение*
 - 5.2.3. Гранулярность параллельных систем
- 5.3. Векторные процессоры
 - 5.3.1. Базовый векторный процессор
 - 5.3.2. Чередующаяся память
 - 5.3.3. Производительность векторного процессора
- 5.4. Матричные процессоры
 - 5.4.1. Базовая организация
 - 5.4.2. Программирование в матричных процессорах
 - 5.4.3. Программирование в матричных процессорах. Наглядный пример
- 5.5. Межсетевые соединения
 - 5.5.1. Межсетевые соединения
 - 5.5.2. Топология, управление потоком и маршрутизация
 - 5.5.3. Межсетевые соединения. Классификация по топологии
- 5.6. Мультипроцессоры
 - 5.6.1. Межсетевые соединения для мультипроцессоров
 - 5.6.2. Согласованность памяти и кэша
 - 5.6.3. Протоколы озвучивания
- 5.7. Синхронизация
 - 5.7.1. Замки (взаимное исключение)
 - 5.7.2. События синхронизации P2P
 - 5.7.3. События глобальной синхронизации
- 5.8. Мультикомпьютеры
 - 5.8.1. Сети мультикомпьютерных соединений
 - 5.8.2. Коммутационный уровень
 - 5.8.3. Уровень маршрутизации

- 5.9. Расширенные архитектуры
 - 5.9.1. Машины потоков данных
 - 5.9.2. Другие архитектуры
- 5.10. Параллельное и распределенное программирование
 - 5.10.1. Языки параллельного программирования
 - 5.10.2. Инструменты параллельного программирования
 - 5.10.3. Паттерны проектирования
 - 5.10.4. Параллельность параллельных и распределенных языков программирования

Модуль 6. Параллельная производительность

- 6.1. Производительность параллельных алгоритмов
 - 6.1.1. Закон Ахмдаля
 - 6.1.2. Закон Густарфсона
 - 6.1.3. Показатели производительности и масштабируемость параллельных алгоритмов
- 6.2. Сравнение параллельных алгоритмов
 - 6.2.1. *Бенчмаркинг*
 - 6.2.2. Математический анализ параллельных алгоритмов
 - 6.2.3. Асимптотический анализ параллельных алгоритмов
- 6.3. Ограничения аппаратных ресурсов
 - 6.3.1. Память
 - 6.3.2. Обработка
 - 6.3.3. Коммуникация
 - 6.3.4. Динамическое разделение ресурсов
- 6.4. Производительность параллельных программ с общей памятью
 - 6.4.1. Оптимальное распределение задач
 - 6.4.2. Средство *потоков*
 - 6.4.3. SIMD-параллелизм
 - 6.4.4. Параллельные программы с общей памятью. Примеры

- 6.5. Производительность параллельных программ с помощью передачи сообщений
 - 6.5.1. Производительность параллельных программ с помощью передачи сообщений
 - 6.5.2. Оптимизация коммуникации в MPI
 - 6.5.3. Контроль сродства и балансировка нагрузки
 - 6.5.4. Параллельный ввод/вывод
 - 6.5.5. Параллельные программы передачи сообщений. Примеры
- 6.6. Производительность гибридных параллельных программ
 - 6.6.1. Производительность гибридных параллельных программ
 - 6.6.2. Гибридное программирование для систем с общей/распределенной памятью
 - 6.6.3. Гибридные параллельные программы. Примеры
- 6.7. Производительность программ с гетерогенными вычислениями
 - 6.7.1. Производительность программ с гетерогенными вычислениями
 - 6.7.2. Гибридное программирование для систем с несколькими аппаратными ускорителями
 - 6.7.3. Программы с гетерогенными вычислениями. Примеры
- 6.8. Анализ производительности параллельных алгоритмов
 - 6.8.1. Анализ производительности параллельных алгоритмов
 - 6.8.2. Анализ производительности параллельных алгоритмов. Инструменты
 - 6.8.3. Анализ производительности параллельных алгоритмов. Рекомендации
- 6.9. Параллельные шаблоны
 - 6.9.1. Параллельные шаблоны
 - 6.9.2. Основные параллельные шаблоны
 - 6.9.3. Параллельные шаблоны Сравнение
- 6.10. Высокопроизводительные параллельные программы
 - 6.10.1. Процесс
 - 6.10.2. Высокопроизводительные параллельные программы
 - 6.10.3. Высокопроизводительные параллельные программы. Реальные применения

Модуль 7. Распределенные вычислительные системы

- 7.1. Распределенные системы
 - 7.1.1. Распределенные системы (PC)
 - 7.1.2. Доказательство теоремы CAP (или гипотезы Брюера)
 - 7.1.3. Ошибки программирования распределенных систем
 - 7.1.4. Повсеместные вычисления
- 7.2. Распределенные системы. Характеристики
 - 7.2.1. Гетерогенность
 - 7.2.2. Расширяемость
 - 7.2.3. Безопасность
 - 7.2.4. Масштабируемость
 - 7.2.5. Устойчивость к сбоям
 - 7.2.6. Параллелизм
 - 7.2.7. Прозрачность
- 7.3. Сети и объединение распределенных сетей
 - 7.3.1. Сети и распределенные системы. Производительность сетей
 - 7.3.2. Сети, доступные для создания распределенной системы. Типология
 - 7.3.3. Протоколы распределенных сетей vs. Централизованных
 - 7.3.4. Взаимосвязь сетей. Интернет
- 7.4. Коммуникация между распределенными процессами
 - 7.4.1. Коммуникация между узлами PC. Проблемы и сбои
 - 7.4.2. Механизмы, реализуемые поверх RPC и RDMA для предотвращения сбоев
 - 7.4.3. Механизмы для программной реализации во избежание сбоев
- 7.5. Проектирование распределенных систем
 - 7.5.1. Эффективное проектирование распределенных систем (PC)
 - 7.5.2. Шаблоны для программирования распределенных систем (PC)
 - 7.5.3. Сервис-ориентированная архитектура (*Service Oriented Architecture (SOA)*)
 - 7.5.4. *Оркестрация сервисов и управление данными микросервисов*

- 7.6. Управление распределенными системами
 - 7.6.1. Мониторинг систем
 - 7.6.2. Внедрение эффективной системы протоколирования (*logging*) в PC
 - 7.6.3. Мониторинг в распределенных сетях
 - 7.6.4. Использование инструмента мониторинга для PC Prometheus и Grafana
- 7.7. Репликация системы
 - 7.7.1. Репликация системы. Типологии
 - 7.7.2. Неизменяемые архитектуры
 - 7.7.3. Контейнерные системы и виртуализация систем как распределенных систем
 - 7.7.4. Сети *блокчейн* как распределенные системы
- 7.8. Распределенные мультимедийные системы
 - 7.8.1. Распределенный обмен изображениями и видео. Проблемы
 - 7.8.2. Серверы мультимедийных объектов
 - 7.8.3. Топология сети для мультимедийной системы
 - 7.8.4. Анализ распределенных мультимедийных систем: Netflix, Amazon, Spotify и т. д.
 - 7.8.5. Распределенные мультимедийные системы в образовании
- 7.9. Распределенные файловые системы
 - 7.9.1. Распределенное совместное использование файлов. Проблемы
 - 7.9.2. Применимость теоремы CAP к базам данных
 - 7.9.3. Распределенные веб-файловые системы: Akamai
 - 7.9.4. Распределенные файловые системы документов IPFS
 - 7.9.5. Распределенные системы баз данных
- 7.10. Подходы к обеспечению безопасности распределенных систем
 - 7.10.1. Безопасность распределенных систем
 - 7.10.2. Известные атаки на распределенные системы
 - 7.10.3. Инструменты для тестирования безопасности PC

Модуль 8. Параллельные вычисления в облачных средах

- 8.1. Облачные вычисления
 - 8.1.1. Современное состояние ИТ-ландшафта
 - 8.1.2. Облако
 - 8.1.3. Облачные вычисления
- 8.2. Безопасность и устойчивость в облаке
 - 8.2.1. Регионы, доступность и зоны сбоев
 - 8.2.2. Управление *арендаторами* или учетными записями в облаке
 - 8.2.3. Управление идентификацией и доступом в облаке
- 8.3. *Сетевое взаимодействие* в облаке
 - 8.3.1. Программно-определяемые виртуальные сети
 - 8.3.2. Сетевые компоненты программно-определяемых сетей
 - 8.3.3. Подключение к другим системам
- 8.4. Облачные сервисы
 - 8.4.1. Инфраструктура как услуга
 - 8.4.2. Платформа как услуга
 - 8.4.3. *Бессерверные* вычисления
 - 8.4.4. Программное обеспечение как услуга
- 8.5. Облачное хранилище
 - 8.5.1. Облачное блочное хранилище
 - 8.5.2. Облачное файловое хранилище
 - 8.5.3. Облачное хранилище объектов
- 8.6. Взаимодействие и мониторинг облака
 - 8.6.1. Мониторинг и управление облаком
 - 8.6.2. Взаимодействие с облаком: консоль администрирования
 - 8.6.3. Взаимодействие с *интерфейсом командной строки*
 - 8.6.4. Взаимодействие на основе API

- 8.7. *Облачно-нативная разработка*
 - 8.7.1. *Облачно-нативная разработка*
 - 8.7.2. *Контейнеры и платформы оркестровки контейнеров*
 - 8.7.3. *Непрерывная облачная интеграция*
 - 8.7.4. *Использование облачных событий*
- 8.8. *Инфраструктура как код в облаке*
 - 8.8.1. *Автоматизация предоставления и управления облаком*
 - 8.8.2. *Terraform*
 - 8.8.3. *Интеграция со сценариями*
- 8.9. *Построение гибридной инфраструктуры*
 - 8.9.1. *Взаимосвязь*
 - 8.9.2. *Взаимосвязь с центром обработки данных*
 - 8.9.3. *Взаимосвязь с другими облаками*
- 8.10. *Высокопроизводительные вычисления*
 - 8.10.1. *Высокопроизводительные вычисления*
 - 8.10.2. *Создание высокопроизводительного кластера*
 - 8.10.3. *Применение высокопроизводительных вычислений*

Модуль 9. Модели и формальная семантика. Программирование, ориентированное на распределенные вычисления

- 9.1. *Семантическая модель данных*
 - 9.1.1. *Семантические модели данных*
 - 9.1.2. *Семантические модели данных. Цели*
 - 9.1.3. *Семантические модели данных. Применение*
- 9.2. *Семантическая модель языков программирования*
 - 9.2.1. *Обработка языка*
 - 9.2.2. *Перевод и интерпретация*
 - 9.2.3. *Гибридные языки*
- 9.3. *Модели вычислений*
 - 9.3.1. *Монолитные вычисления*
 - 9.3.2. *Параллельные вычисления*
 - 9.3.3. *Распределенные вычисления*
 - 9.3.4. *Кооперативные вычисления (P2P)*

- 9.4. *Параллельные вычисления*
 - 9.4.1. *Параллельная архитектура*
 - 9.4.2. *Аппаратное обеспечение*
 - 9.4.3. *Программное обеспечение*
- 9.5. *Распределенная модель. Грид-вычисления*
 - 9.5.1. *Архитектура грид-вычислений*
 - 9.5.2. *Архитектура грид-вычислений. Анализ*
 - 9.5.3. *Архитектура грид-вычислений. Применение*
- 9.6. *Распределенная модель. Кластерные вычисления*
 - 9.6.1. *Архитектура кластерных вычислений*
 - 9.6.2. *Архитектура кластерных вычислений. Анализ*
 - 9.6.3. *Архитектура кластерных вычислений. Применение*
- 9.7. *Кластерные вычисления. Современные инструменты для их реализации. Гипервизоры*
 - 9.7.1. *Конкуренты на рынке*
 - 9.7.2. *Гипервизор VMware*
 - 9.7.3. *Hyper-V*
- 9.8. *Распределенная модель. Облачные вычисления*
 - 9.8.1. *Архитектура облачных вычислений*
 - 9.8.2. *Архитектура облачных вычислений. Анализ*
 - 9.8.3. *Архитектура облачных вычислений. Применение*
- 9.9. *Распределенная модель. Облачные вычисления Amazon*
 - 9.9.1. *Облачные вычисления Amazon. Функции*
 - 9.9.2. *Облачные вычисления Amazon. Лицензирование*
 - 9.9.3. *Облачные вычисления Amazon. Эталонные архитектуры*
- 9.10. *Распределенная модель. Облачные вычисления Microsoft*
 - 9.10.1. *Облачные вычисления Microsoft. Функции*
 - 9.10.2. *Облачные вычисления Microsoft. Лицензирование*
 - 9.10.3. *Облачные вычисления Microsoft. Эталонные архитектуры*

Модуль 10. Применение параллельных и распределенных вычислений

- 10.1. Параллельные и распределенные вычисления в современных приложениях
 - 10.1.1. Аппаратное обеспечение
 - 10.1.2. Программное обеспечение
 - 10.1.3. Важность синхронизации
- 10.2. Климат. Климатические изменения
 - 10.2.1. Климатические приложения. Источники данных
 - 10.2.2. Климатические приложения. Объемы данных
 - 10.2.3. Климатические приложения. В реальном времени
- 10.3. Параллельные вычисления на GPU
 - 10.3.1. Параллельные вычисления на GPU
 - 10.3.2. GPUs vs. CPU. Использование GPU
 - 10.3.3. GPU Примеры
- 10.4. Интеллектуальный *грид*. Вычисления в электросетях
 - 10.4.1. Интеллектуальный *грид*
 - 10.4.2. Концептуальные модели. Примеры
 - 10.4.3. Интеллектуальный *грид*. Пример
- 10.5. Распределенный двигатель. *ElasticSearch*
 - 10.5.1. Распределенный двигатель. *ElasticSearch*
 - 10.5.2. Архитектура с *ElasticSearch*. Примеры
 - 10.5.3. Распределенный двигатель. Примеры использования
- 10.6. *Фреймворк больших данных*
 - 10.6.1. *Фреймворк больших данных*
 - 10.6.2. Архитектура расширенных инструментов
 - 10.6.3. *Большие данные* в распределенных вычислениях
- 10.7. База данных в памяти
 - 10.7.1. База данных в памяти
 - 10.7.2. Решение Redis. Случай успеха
 - 10.7.3. Развертывание решений для баз данных в памяти

- 10.8. *Блокчейн*
 - 10.8.1. Архитектура *блокчейна*. Компоненты
 - 10.8.2. Взаимодействие между узлами и консенсус
 - 10.8.3. Решения *блокчейна*. Способы реализации
- 10.9. Распределенные системы в медицине
 - 10.9.1. Компоненты архитектуры
 - 10.9.2. Распределенные системы в медицине. Как это работает
 - 10.9.3. Распределенные системы в медицине. Применение
- 10.10. Распределенные системы в авиации
 - 10.10.1. Проектирование архитектуры
 - 10.10.2. Распределенные системы в авиационном секторе. Функциональные возможности компонентов
 - 10.10.3. Распределенные системы в авиационном секторе Применение



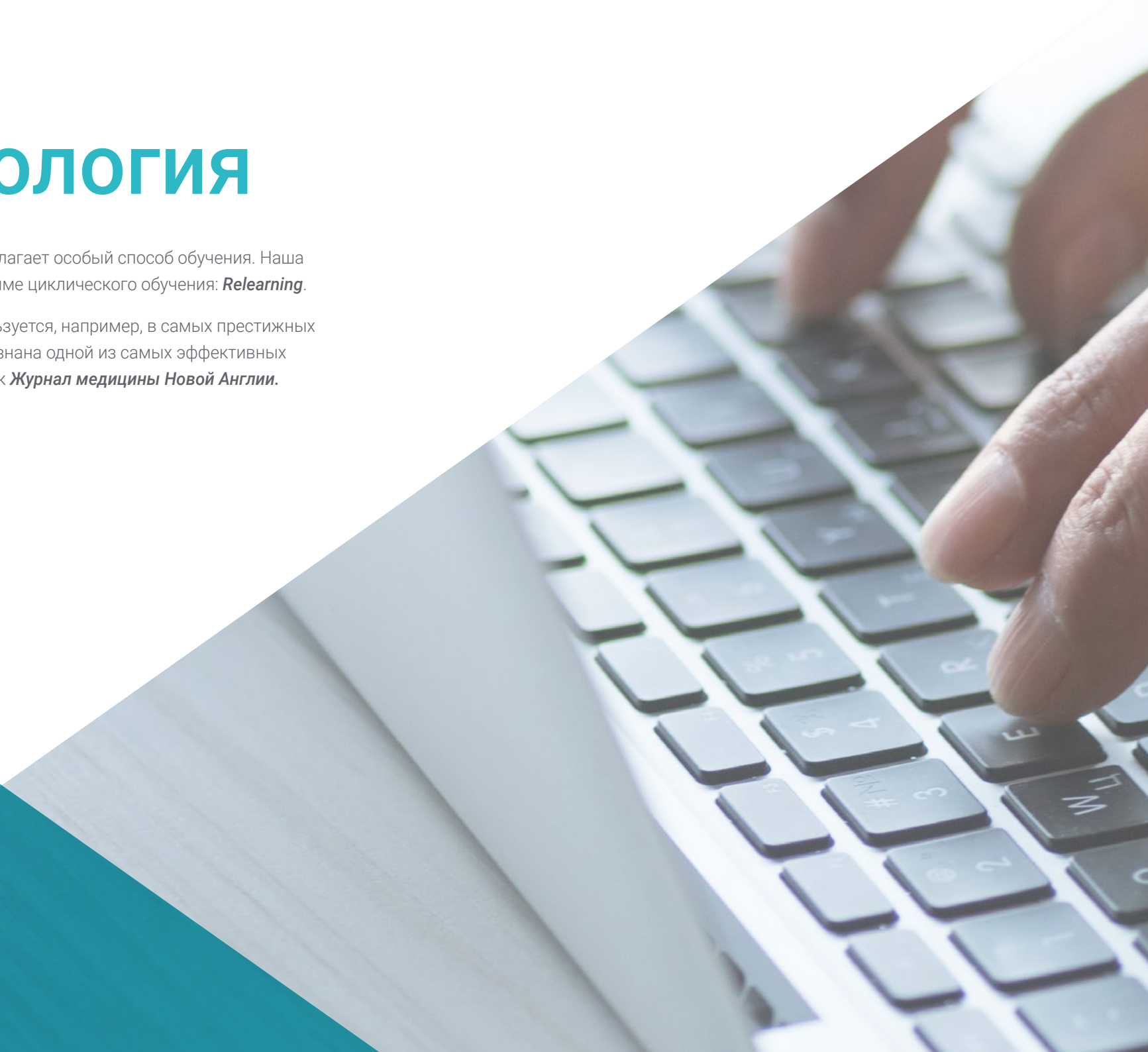
Данная программа — это толчок, необходимый вам для достижения профессионального роста, которого вы заслуживаете и ради которого работали"

06

Методология

Данная учебная программа предлагает особый способ обучения. Наша методология разработана в режиме циклического обучения: **Relearning**.

Данная система обучения используется, например, в самых престижных медицинских школах мира и признана одной из самых эффективных ведущими изданиями, такими как **Журнал медицины Новой Англии**.



“

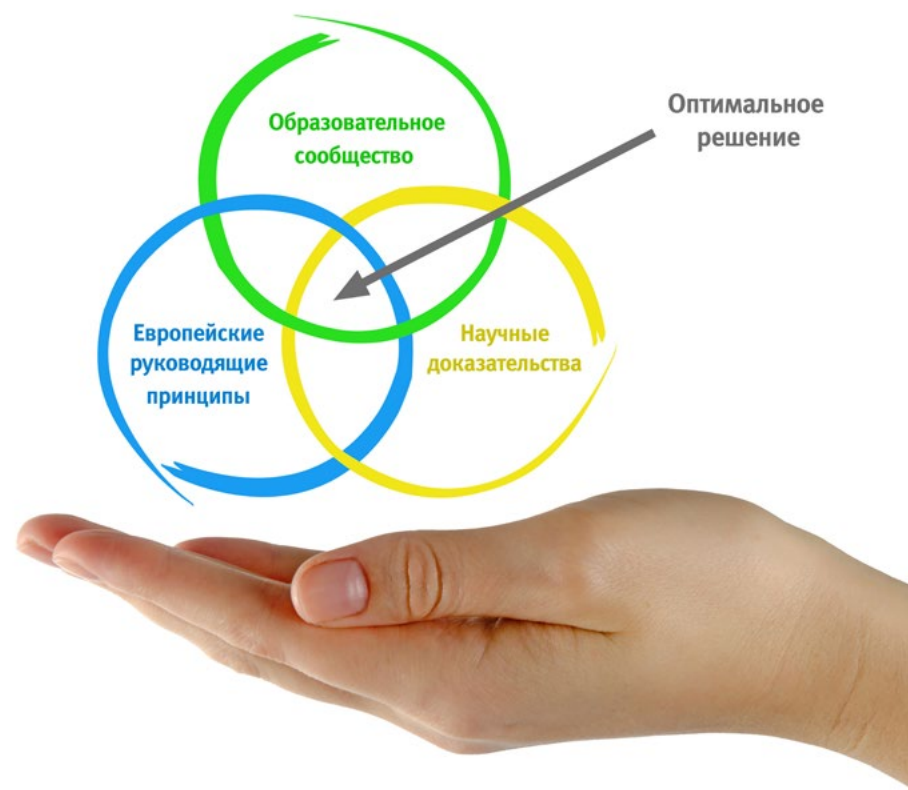
Откройте для себя методику *Relearning*, которая отвергает традиционное линейное обучение, чтобы показать вам циклические системы обучения: способ, который доказал свою огромную эффективность, особенно в предметах, требующих запоминания”

Исследование кейсов для контекстуализации всего содержания

Наша программа предлагает революционный метод развития навыков и знаний. Наша цель - укрепить компетенции в условиях меняющейся среды, конкуренции и высоких требований.

“

С TESH вы сможете познакомиться со способом обучения, который опровергает основы традиционных методов образования в университетах по всему миру”



Вы получите доступ к системе обучения, основанной на повторении, с естественным и прогрессивным обучением по всему учебному плану.



В ходе совместной деятельности и рассмотрения реальных кейсов студент научится разрешать сложные ситуации в реальной бизнес-среде.

Инновационный и отличный от других метод обучения

Эта программа TECH - интенсивная программа обучения, созданная с нуля, которая предлагает самые сложные задачи и решения в этой области на международном уровне. Благодаря этой методологии ускоряется личностный и профессиональный рост, делая решающий шаг на пути к успеху. Метод кейсов, составляющий основу данного содержания, обеспечивает следование самым современным экономическим, социальным и профессиональным реалиям.



Наша программа готовит вас к решению новых задач в условиях неопределенности и достижению успеха в карьере"

Кейс-метод является наиболее широко используемой системой обучения лучшими преподавателями в мире. Разработанный в 1912 году для того, чтобы студенты-юристы могли изучать право не только на основе теоретического содержания, метод кейсов заключается в том, что им представляются реальные сложные ситуации для принятия обоснованных решений и ценностных суждений о том, как их разрешить. В 1924 году он был установлен в качестве стандартного метода обучения в Гарвардском университете.

Что должен делать профессионал в определенной ситуации? Именно с этим вопросом мы сталкиваемся при использовании кейс-метода - метода обучения, ориентированного на действие. На протяжении всей курса студенты будут сталкиваться с многочисленными реальными случаями из жизни. Им придется интегрировать все свои знания, исследовать, аргументировать и защищать свои идеи и решения.

Методология *Relearning*

TECH эффективно объединяет метод кейсов с системой 100% онлайн-обучения, основанной на повторении, которая сочетает различные дидактические элементы в каждом уроке.

Мы улучшаем метод кейсов с помощью лучшего метода 100% онлайн-обучения: *Relearning*.

В 2019 году мы достигли лучших результатов обучения среди всех онлайн-университетов в мире.

В TECH вы будете учиться по передовой методике, разработанной для подготовки руководителей будущего. Этот метод, играющий ведущую роль в мировой педагогике, называется *Relearning*.

Наш университет - единственный вуз, имеющий лицензию на использование этого успешного метода. В 2019 году нам удалось повысить общий уровень удовлетворенности наших студентов (качество преподавания, качество материалов, структура курса, цели...) по отношению к показателям лучшего онлайн-университета.





В нашей программе обучение не является линейным процессом, а происходит по спирали (мы учимся, разучиваемся, забываем и заново учимся). Поэтому мы дополняем каждый из этих элементов по концентрическому принципу. Благодаря этой методике более 650 000 выпускников университетов добились беспрецедентного успеха в таких разных областях, как биохимия, генетика, хирургия, международное право, управленческие навыки, спортивная наука, философия, право, инженерное дело, журналистика, история, финансовые рынки и инструменты. Наша методология преподавания разработана в среде с высокими требованиями к уровню подготовки, с университетским контингентом студентов с высоким социально-экономическим уровнем и средним возрастом 43,5 года.

Методика Relearning позволит вам учиться с меньшими усилиями и большей эффективностью, все больше вовлекая вас в процесс обучения, развивая критическое мышление, отстаивая аргументы и противопоставляя мнения, что непосредственно приведет к успеху.

Согласно последним научным данным в области нейронауки, мы не только знаем, как организовать информацию, идеи, образы и воспоминания, но и знаем, что место и контекст, в котором мы что-то узнали, имеют фундаментальное значение для нашей способности запомнить это и сохранить в гиппокампе, чтобы удержать в долгосрочной памяти.

Таким образом, в рамках так называемого нейрокогнитивного контекстно-зависимого электронного обучения, различные элементы нашей программы связаны с контекстом, в котором участник развивает свою профессиональную практику.

В рамках этой программы вы получаете доступ к лучшим учебным материалам, подготовленным специально для вас:



Учебный материал

Все дидактические материалы создаются преподавателями специально для студентов этого курса, чтобы они были действительно четко сформулированными и полезными.

Затем вся информация переводится в аудиовизуальный формат, создавая дистанционный рабочий метод TECH. Все это осуществляется с применением новейших технологий, обеспечивающих высокое качество каждого из представленных материалов.



Мастер-классы

Существуют научные данные о пользе экспертного наблюдения третьей стороны.

Так называемый метод обучения у эксперта укрепляет знания и память, а также формирует уверенность в наших будущих сложных решениях.



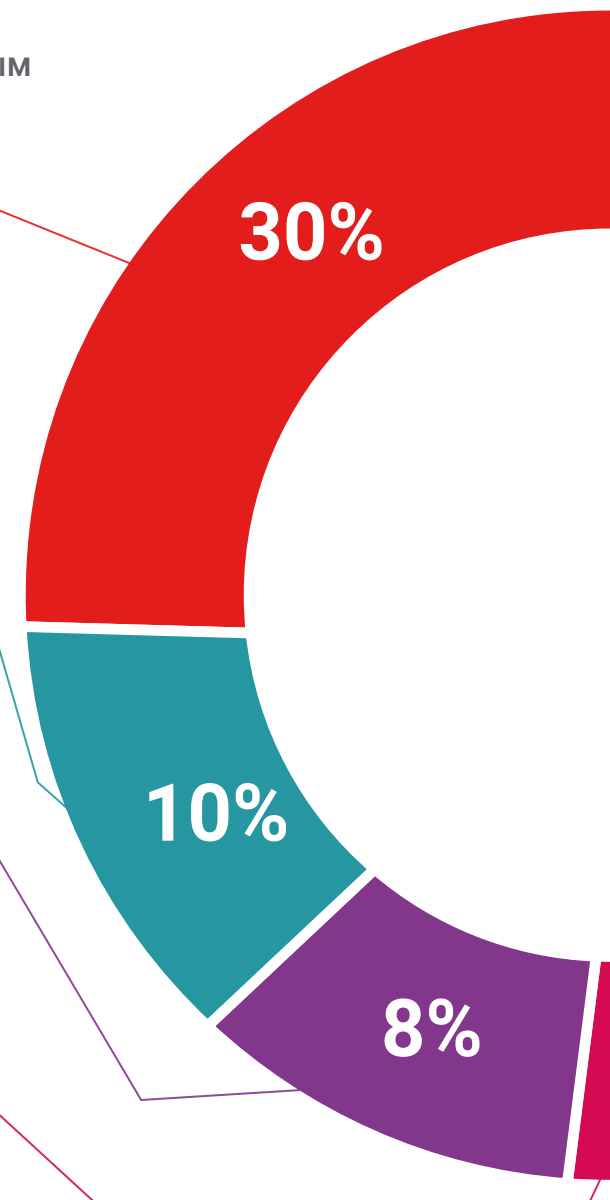
Практика навыков и компетенций

Студенты будут осуществлять деятельность по развитию конкретных компетенций и навыков в каждой предметной области. Практика и динамика приобретения и развития навыков и способностей, необходимых специалисту в рамках глобализации, в которой мы живем.



Дополнительная литература

Новейшие статьи, консенсусные документы и международные руководства включены в список литературы курса. В виртуальной библиотеке TECH студент будет иметь доступ ко всем материалам, необходимым для завершения обучения.





Метод кейсов

Метод дополнится подборкой лучших кейсов, выбранных специально для этой квалификации. Кейсы представляются, анализируются и преподаются лучшими специалистами на международной арене.



Интерактивные конспекты

Мы представляем содержание в привлекательной и динамичной мультимедийной форме, которая включает аудио, видео, изображения, диаграммы и концептуальные карты для закрепления знаний. Эта уникальная обучающая система для представления мультимедийного содержания была отмечена компанией Microsoft как "Европейская история успеха".



Тестирование и повторное тестирование

На протяжении всей программы мы периодически оцениваем и переоцениваем ваши знания с помощью оценочных и самооценочных упражнений: так вы сможете убедиться, что достигаете поставленных целей.



07

Квалификация

Специализированная магистратура в области Параллельные и распределенные вычисления гарантирует, помимо самого строгого и современного обучения, получение диплома об окончании Специализированной магистратуры, выдаваемого ТЕСН Технологическим университетом.



“

Успешно пройдите эту программу и получите университетский диплом без хлопот, связанных с поездками и оформлением документов”

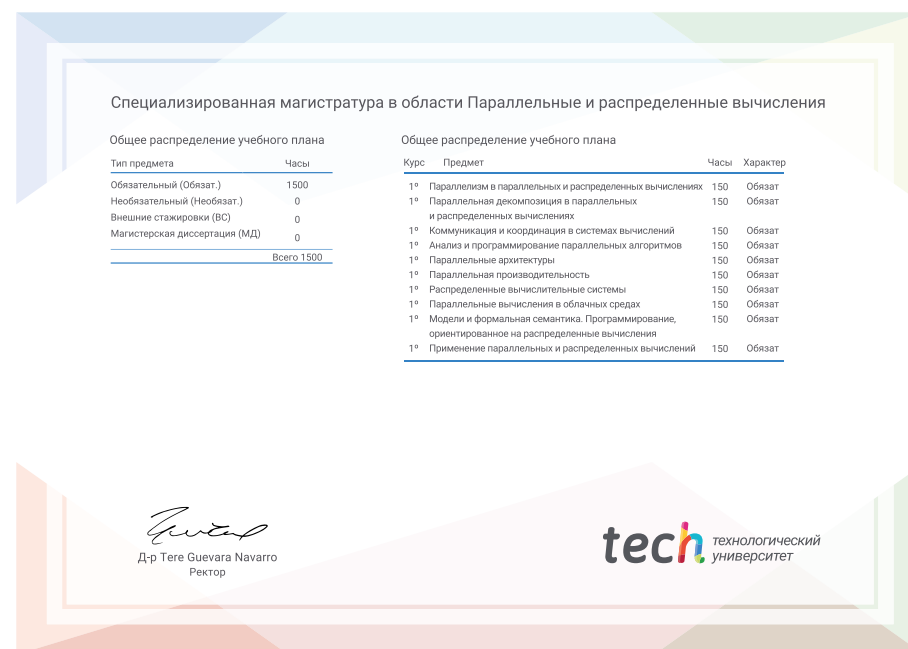
Данная **Специализированная магистратура в области Параллельные и распределенные вычисления** содержит самую полную и современную программу на рынке.

После прохождения аттестации студент получит по почте* с подтверждением получения соответствующий диплом **Специализированной магистратуры**, выданный **TECH Технологическим университетом**.

Диплом, выданный **TECH Технологическим университетом**, подтверждает квалификацию, полученную в Специализированной магистратуре, и соответствует требованиям, обычно предъявляемым биржами труда, конкурсными экзаменами и комитетами по оценке карьеры.

Диплом: **Специализированная магистратура в области Параллельные и распределенные вычисления**

Количество учебных часов: **1500 часов**



*Гаагский апостиль. В случае, если студент потребует, чтобы на его диплом в бумажном формате был проставлен Гаагский апостиль, TECH EDUCATION предпримет необходимые шаги для его получения за дополнительную плату.

Будущее

Здоровье Доверие Люди

Образование Информация Тьюторы

Гарантия Аккредитация Преподавание

Институты Технология Обучение

Сообщество Обязательства

Персональное внимание Инновации

Знания Настоящее Качество

Веб обучение

Развитие Институты

Виртуальный класс Языки

tech технологический
университет

Специализированная
магистратура

Параллельные и распределенные
вычисления

- » Формат: онлайн
- » Продолжительность: 12 месяцев
- » Учебное заведение: TESH Технологический университет
- » Режим обучения: 16ч./неделя
- » Расписание: по своему усмотрению
- » Экзамены: онлайн

Специализированная магистратура Параллельные и распределенные вычисления

