

# Privater Masterstudiengang

## Parallele und Verteilte Datenverarbeitung





## Privater Masterstudiengang Parallele und Verteilte Datenverarbeitung

- » Modalität: **online**
- » Dauer: **12 Monate**
- » Qualifizierung: **TECH Technische Universität**
- » Aufwand: **16 Std./Woche**
- » Zeitplan: **in Ihrem eigenen Tempo**
- » Prüfungen: **online**

Internetzugang: [www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-parallele-verteilte-datenverarbeitung](http://www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-parallele-verteilte-datenverarbeitung)

# Index

01

Präsentation

---

Seite 4

02

Ziele

---

Seite 8

03

Kompetenzen

---

Seite 14

04

Kursleitung

---

Seite 18

05

Struktur und Inhalt

---

Seite 22

06

Methodik

---

Seite 32

07

Qualifizierung

---

Seite 40

# 01

# Präsentation

Die meisten Software- und Elektroniksysteme nutzen heute in der einen oder anderen Form parallele oder verteilte Datenverarbeitung. Beispielsweise wurden Smartphones im Laufe der Jahre mit immer leistungsfähigeren Mehrkernprozessoren ausgestattet, und in modernen Computern sind Mehrkernprozessoren inzwischen gang und gäbe. Auf der anderen Seite hat die verteilte Datenverarbeitung andere Forschungsbereiche wie *Big Data* vorangetrieben und ist in Bereichen wie soziale Netzwerke, Unternehmensnetzwerke oder Multiplayer-Onlinespiele unverzichtbar. All dies zeigt, wie wichtig diese beiden Arten des Programmierens sind. Aus diesem Grund hat TECH dieses komplette Universitätsprogramm entwickelt, in dem Informatiker die Vorteile und die wichtigsten Anwendungen der parallelen und verteilten Datenverarbeitung kennen lernen.







“

*Verbessern Sie Ihre Karriere und Ihren Lebenslauf, indem Sie die neuesten Kenntnisse über parallele und verteilte Datenverarbeitung in Ihre Arbeit integrieren”*

Gute, fortgeschrittene Kenntnisse in der parallelen und verteilten Datenverarbeitung können der Karriere eines jeden Informatikers, der sich profilieren möchte, Auftrieb geben. Da es sich um ein komplexes Thema handelt, das zu einer Vielzahl von Anwendungen führen kann, hat TECH ein Team von Experten auf diesem Gebiet mit der Ausarbeitung aller Inhalte beauftragt.

So finden Informatiker unter anderem Themen wie Kommunikation und Koordination in Computersystemen, Analyse und Programmierung paralleler Algorithmen oder verteilte Systeme in der Informatik. All dies wird aus einer modernen und innovativen Perspektive geschrieben, die auf der gesammelten Erfahrung der Dozenten selbst basiert.

Informatiker, die dieses Programm absolvieren, haben somit einen entscheidenden Vorteil bei der Planung ihrer Karriere in Richtung der Entwicklung von Anwendungen oder Systemen in den Bereichen Klima, Gesundheit, *Big Data*, *Cloud Computing* oder *Blockchain*. Aufgrund des fortgeschrittenen Charakters des Studiengangs ist es sogar möglich, eine Forschungslaufbahn in der Informatik oder in anderen verwandten Bereichen anzustreben.

Darüber hinaus wird der Studiengang vollständig online angeboten, so dass keine Anwesenheitspflicht besteht und die typischen Zwänge eines vorgegebenen Stundenplans entfallen. Informatiker können ihr Studienpensum frei nach ihren eigenen Interessen einteilen und den privaten Masterstudiengang mit anderen persönlichen oder beruflichen Verpflichtungen kombinieren.

Dieser **Privater Masterstudiengang in Parallele und Verteilte Datenverarbeitung** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- ◆ Die Entwicklung von Fallstudien, die von Experten für parallele und verteilte Datenverarbeitung vorgestellt werden
- ◆ Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren Informationen
- ◆ Die praktischen Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens durchgeführt werden kann
- ◆ Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- ◆ Theoretischer Unterricht, Fragen an den Experten und individuelle Reflexionsarbeit
- ◆ Die Verfügbarkeit des Zugangs zu Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss



*Schreiben Sie sich jetzt ein und entdecken Sie die neuesten Entwicklungen im Bereich der parallelen Datenverarbeitung in Cloud-Umgebungen und der auf verteilte Datenverarbeitung ausgerichteten Programmierung“*

“

*Sie werden jederzeit von einem Dozententeam beraten, das sich aus Fachleuten mit umfangreicher Erfahrung im Bereich der parallelen und verteilten Datenverarbeitung zusammensetzt“*

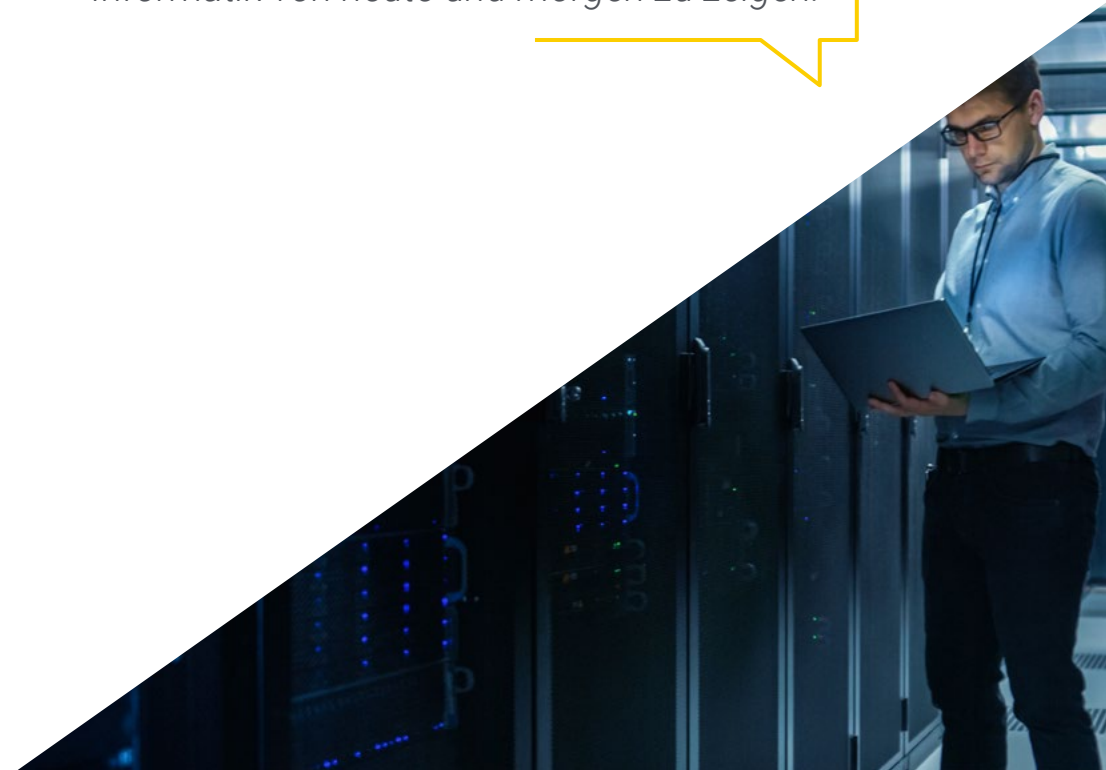
Zu den Dozenten des Programms gehören Experten aus der Branche, die ihre Berufserfahrung in diese Fortbildung einbringen, sowie renommierte Fachleute von Referenzgesellschaften und angesehenen Universitäten.

Die multimedialen Inhalte, die mit der neuesten Bildungstechnologie entwickelt wurden, werden den Fachkräften ein situierendes und kontextbezogenes Lernen ermöglichen, d. h. eine simulierte Umgebung, die eine immersive Fortbildung bietet, die auf die Ausführung von realen Situationen ausgerichtet ist.

Das Konzept dieses Programms konzentriert sich auf problemorientiertes Lernen, bei dem die Fachkräfte versuchen müssen, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Zu diesem Zweck werden sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

*Sie erhalten die volle Unterstützung von TECH, der weltweit größten akademischen Online-Institution, die Ihnen die neuesten Bildungstechnologien zur Verfügung stellt.*

*Verpassen Sie nicht die Gelegenheit, sich zu profilieren und Ihr Interesse an der Informatik von heute und morgen zu zeigen.*





# 02 Ziele

In Anbetracht der Tatsache, dass sich die Informatik rasant entwickelt und dass Fachleute auf diesem Gebiet ihr Wissen ständig aktualisieren müssen, hat TECH dieses Programm auf die neuesten Entwicklungen im Bereich der parallelen und verteilten Datenverarbeitung ausgerichtet. Auf diese Weise erwerben die Studenten nicht nur die fortschrittlichsten Fähigkeiten in diesem Bereich, sondern lernen auch die vielfältigen Anwendungen kennen, die Technologien wie *Blockchain* oder *Cloud Computing* heute bieten.





“

*Mit den Tipps und Tricks, die Sie in diesem Programm lernen, kommen Sie Ihrem Ziel, sich beruflich weiterzuentwickeln, einen großen Schritt näher”*



## Allgemeine Ziele

---

- ◆ Analysieren, was zwischen den verschiedenen Komponenten der parallelen und verteilten Datenverarbeitung geschieht
- ◆ Messen und Vergleichen deren Leistung, um die Leistung der verwendeten Komponenten zu analysieren
- ◆ Eingehendes Analysieren der plattformübergreifenden parallelen Datenverarbeitung zur Nutzung von Parallelität auf Aufgabenebene zwischen verschiedenen Hardwarebeschleunigern
- ◆ Analysieren der aktuellen Software und Architekturen im Detail
- ◆ Vertiefen der relevanten Aspekte der parallelen und verteilten Datenverarbeitung
- ◆ Spezialisieren der Studenten auf den Einsatz von paralleler und verteilter Datenverarbeitung in verschiedenen Anwendungsbereichen



*Sie erhalten einen umfassenden Einblick in alle wichtigen Themen der parallelen und verteilten Datenverarbeitung, von der Parallelität selbst bis hin zu ihren vielfältigen Anwendungen"*



## Spezifische Ziele

---

### **Modul 1. Parallelität in der parallelen und verteilten Datenverarbeitung**

- ◆ Analysieren der Verarbeitungskomponenten: Prozessor oder Speicher
- ◆ Vertiefen in die Architektur der Parallelität
- ◆ Analysieren der verschiedenen Formen der Parallelität aus der Sicht des Prozessors

### **Modul 2. Parallele Zerlegung in der parallelen und verteilten Datenverarbeitung**

- ◆ Analysieren der Bedeutung der parallelen Prozesszerlegung bei der Lösung von Rechenproblemen
- ◆ Untersuchen verschiedener Beispiele zur Demonstration der Anwendung und Nutzung von Datenverarbeitung und ihrer parallelen Zerlegung
- ◆ Entdecken von Verfahren und Werkzeugen, die die Ausführung von parallelen Prozessen ermöglichen, um die bestmögliche Leistung zu erzielen
- ◆ Entwickeln von Fachwissen, um Szenarien für die Zerlegung paralleler Prozesse zu identifizieren und das geeignete Werkzeug auszuwählen und anzuwenden

### **Modul 3. Kommunikation und Koordination in Computersystemen**

- ◆ Analysieren der verschiedenen Architekturen und Modelle von verteilten Systemen
- ◆ Bestimmen der Eigenschaften von parallelen und verteilten Systemen
- ◆ Eingehendes Untersuchen der verschiedenen Kommunikationen, die auf der Prozessebene stattfinden
- ◆ Untersuchen der Remote-, flussorientierten, nachrichtenorientierten und Multicast-Kommunikation zusammen mit neueren Beispielen und Überlegungen
- ◆ Ermitteln der sich entwickelnden Kommunikationstypen, ihrer Stärken und Grenzen
- ◆ Entwickeln der Prozesse, die bei der Wahl der Algorithmen für den Namensdienst, die Uhrensynchronisation, die Koordination und die Vereinbarung zwischen den Elementen des Systems zu beachten sind
- ◆ Zusammenstellen von Szenarien, in denen verschiedene Arten von Kommunikationstechnologien eingesetzt werden, die die Leistung und Skalierbarkeit verbessern

#### Modul 4. Analyse und Programmierung von parallelen Algorithmen

- ◆ Analysieren der verschiedenen Paradigmen der parallelen Programmierung
- ◆ Untersuchen der modernsten Tools für die Durchführung der parallelen Programmierung
- ◆ Analysieren paralleler Algorithmen für grundlegende Probleme
- ◆ Entwerfen und Analysieren von parallelen Algorithmen
- ◆ Entwickeln paralleler Algorithmen und deren Implementierung mit MPI, OpenMP, OpenCL/CUDA

#### Modul 5. Parallele Architekturen

- ◆ Analysieren der wichtigsten Computerarchitekturen
- ◆ Vertiefen von Schlüsselaspekten wie Prozess, Dienst und Ausführungsfaden
- ◆ Verwalten von laufenden Prozessen in einem Betriebssystem
- ◆ Verwenden von Klassen zum Starten und Verwalten von Prozessen

#### Modul 6. Parallele Leistung

- ◆ Analysieren der Aspekte paralleler Algorithmen, die deren Leistung und Skalierbarkeit beeinflussen
- ◆ Ermitteln der wichtigsten Leistungs- und Skalierbarkeitskennzahlen von parallelen Algorithmen
- ◆ Untersuchen der wichtigsten Benchmarking-Techniken für parallele Algorithmen
- ◆ Identifizieren der Einschränkungen, die die Hardware-Ressourcen für die Parallelisierung mit sich bringen

- ◆ Ermitteln von *Best Practices* für die Leistung von parallelen Programmen mit gemeinsamem Speicher, für die Leistung von parallelen Programmen mit Message-Passing, für die Leistung von hybriden parallelen Programmen und für die Leistung von parallelen Programmen mit heterogener Datenverarbeitung
- ◆ Zusammenstellen modernster Tools zur Analyse der Leistung paralleler Algorithmen
- ◆ Einführen der wichtigsten Muster der parallelen Verarbeitung
- ◆ Spezifizieren eines robusten Verfahrens für die Definition von leistungsstarken parallelen Programmen

#### Modul 7. Verteilte Systeme in der Datenverarbeitung

- ◆ Entwickeln der Schlüsselemente eines verteilten Systems
- ◆ Untersuchen der Sicherheitselemente, die in verteilten Systemen eingesetzt werden, und der Notwendigkeit dieser Elemente
- ◆ Vorstellen der verschiedenen Arten von verteilten Systemen, die am häufigsten verwendet werden, sowie deren Eigenschaften, Funktionalitäten und zu lösende Probleme
- ◆ Demonstration des CAP-Theorems, das auf verteilte Systeme anwendbar ist: *Consistency* (Konsistenz), *Availability* (Verfügbarkeit) und *Partition Tolerance* (Partitionstoleranz)

#### Modul 8. Parallele Datenverarbeitung angewandt auf Cloud-Umgebungen

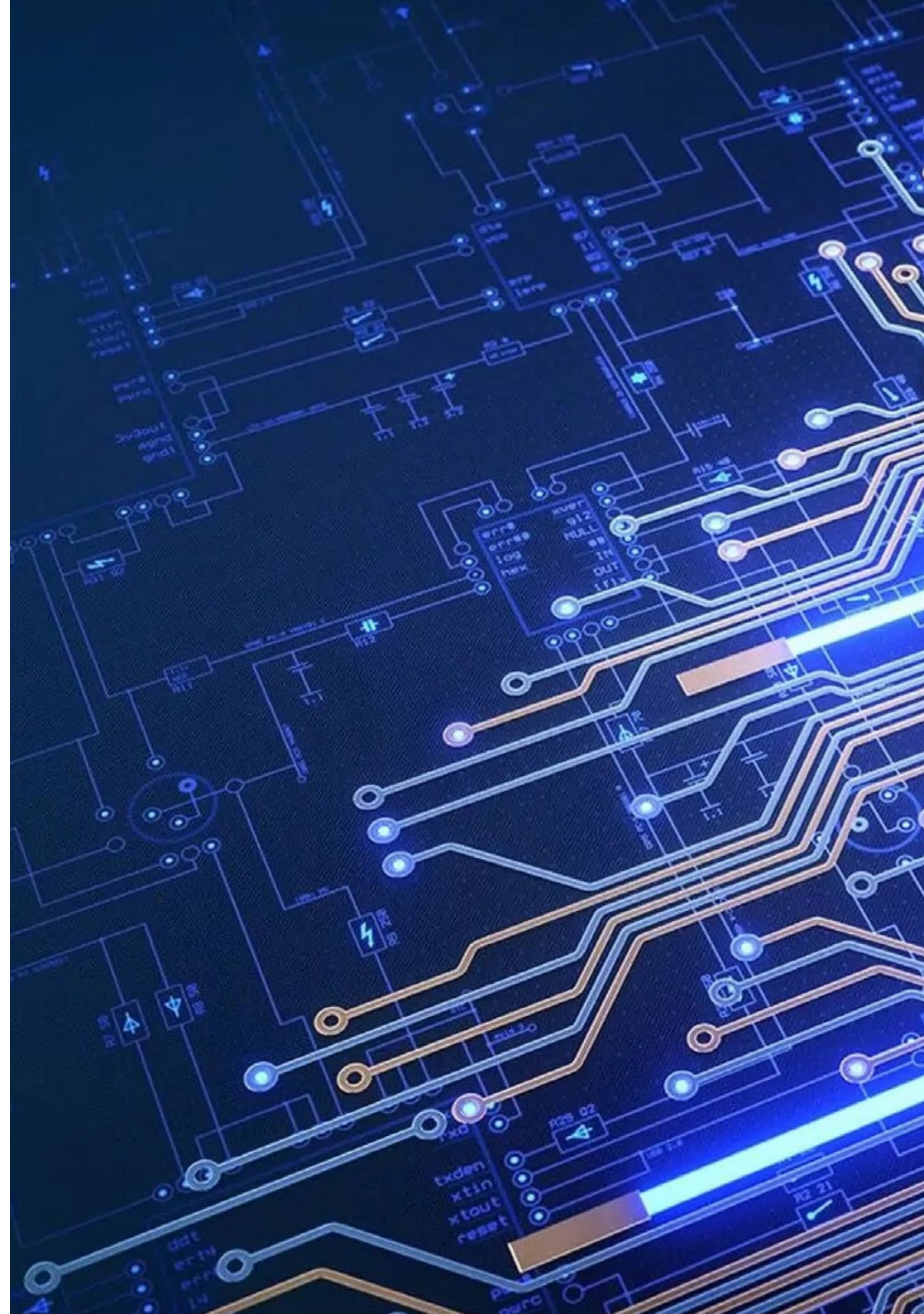
- ◆ Entwickeln des Paradigmas des *Cloud Computing*
- ◆ Identifizieren der verschiedenen Ansätze auf der Grundlage des Automatisierungs- und Servicegrads
- ◆ Analysieren der Hauptbestandteile einer Cloud-Architektur
- ◆ Ermitteln der Unterschiede zu einer *On-Premise*-Architektur
- ◆ Analysieren der verschiedenen Cloud-Bereitstellungsoptionen: *Multi-Cloud*, *Hybrid Cloud*
- ◆ Vertiefen der inhärenten Vorteile von *Cloud Computing*



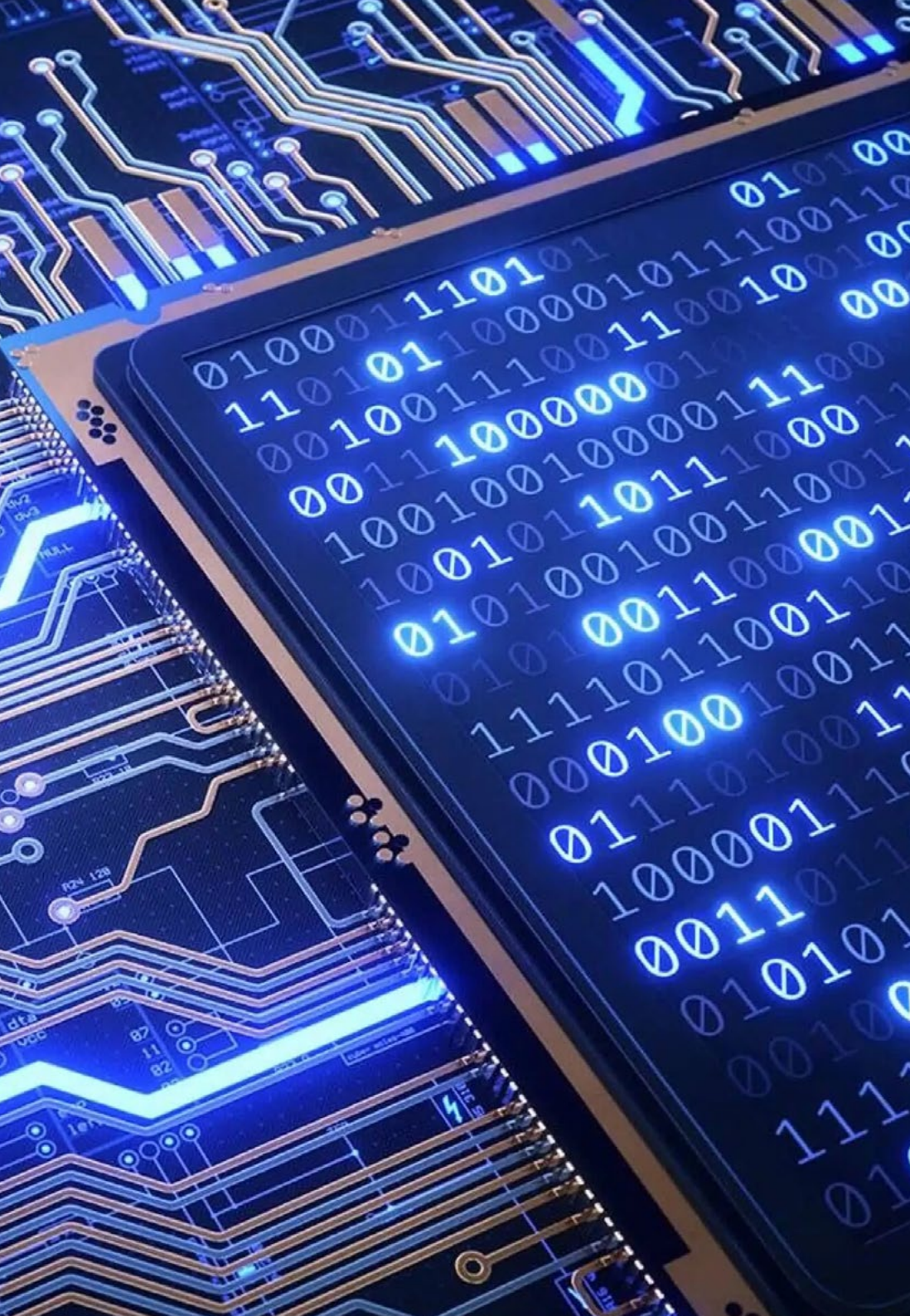
- ◆ Entwickeln der Grundsätze der Wirtschaftlichkeit von *Cloud Computing*: Wechsel von CAPEX zu OPEX
- ◆ Bewerten der kommerziellen Angebote der verschiedenen Cloud-Anbieter
- ◆ Bewerten der Möglichkeiten von Supercomputing in der Cloud
- ◆ Untersuchen der Sicherheit beim *Cloud Computing*

### Modul 9. Modelle und formale Semantik. Programmierung für verteilte Datenverarbeitung

- ◆ Identifizieren der Vorteile der formalen Semantik
- ◆ Untersuchen, wie formale Semantik bei der verteilten rechnergestützten Programmierung hilft
- ◆ Konkretisieren der Möglichkeiten der formalen Semantik bei der Anwendung auf die verteilte rechnergestützte Datenverarbeitung
- ◆ Entwickeln der wichtigsten Hilfsmittel für die Durchführbarkeit von Projekten, die diese Technologie nutzen, im Detail
- ◆ Identifizieren von Programmiersprachen im semantischen Modell
- ◆ Ermitteln, wie diese semantischen Modelle uns bei Programmiersprachen helfen
- ◆ Bewerten und Vergleichen von Datenverarbeitungsmodellen
- ◆ Konkretisieren des Einsatzes von verteilten Modellen
- ◆ Vorstellen der fortschrittlichsten Marktwerkzeuge für Projekte







### Modul 10. Anwendungen der parallelen und verteilten Datenverarbeitung

- ◆ Aufzeigen des großen Beitrags von Anwendungen der parallelen und verteilten Datenverarbeitung für unsere Umwelt
- ◆ Bestimmen der Referenzarchitekturen auf dem Markt
- ◆ Bewerten der Vorteile dieser Anwendungsfälle
- ◆ Präsentieren erfolgreicher Lösungen auf dem Markt
- ◆ Aufzeigen, warum sie für die Bewertung des Klimawandels wichtig sind
- ◆ Bestimmen der aktuellen Bedeutung von GPUs
- ◆ Präsentieren der Auswirkungen dieser Technologie auf die Stromnetze
- ◆ Erforschen von verteilten Motoren im Dienste unserer Kunden
- ◆ Verstehen, welche Vorteile verteilte Engines für unsere Unternehmen bringen
- ◆ Präsentieren von Beispielen für *In-Memory*-Datenbanken und deren Bedeutung
- ◆ Untersuchen, wie diese Modelle der Medizin helfen



# 03

# Kompetenzen

Mit den richtigen Kenntnissen in diesem Bereich der Informatik können sich Fachleute auszeichnen und an die Spitze ganzer Programmiererteams gestellt werden. Ein Beweis dafür sind die Dozenten selbst, die in verantwortungsvollen Positionen und an der Spitze vieler verschiedener Informatikprojekte hervorragende Erfahrungen gesammelt haben. Aus diesem Grund konzentriert sich der Private Masterstudiengang in Parallele und Verteilte Datenverarbeitung nicht nur auf die wesentlichen Schlüssel dieses Fachgebiets, sondern vermittelt dem Informatiker auch eine Reihe unverwechselbarer und einzigartiger Kompetenzen, die ihn in seiner täglichen Arbeit auszeichnen werden.





“

*Diese Qualifikation ist der Schlüssel zu einem kompetentem Aufstieg an die Spitze der Programmierung und Entwicklung aller Arten von Software”*





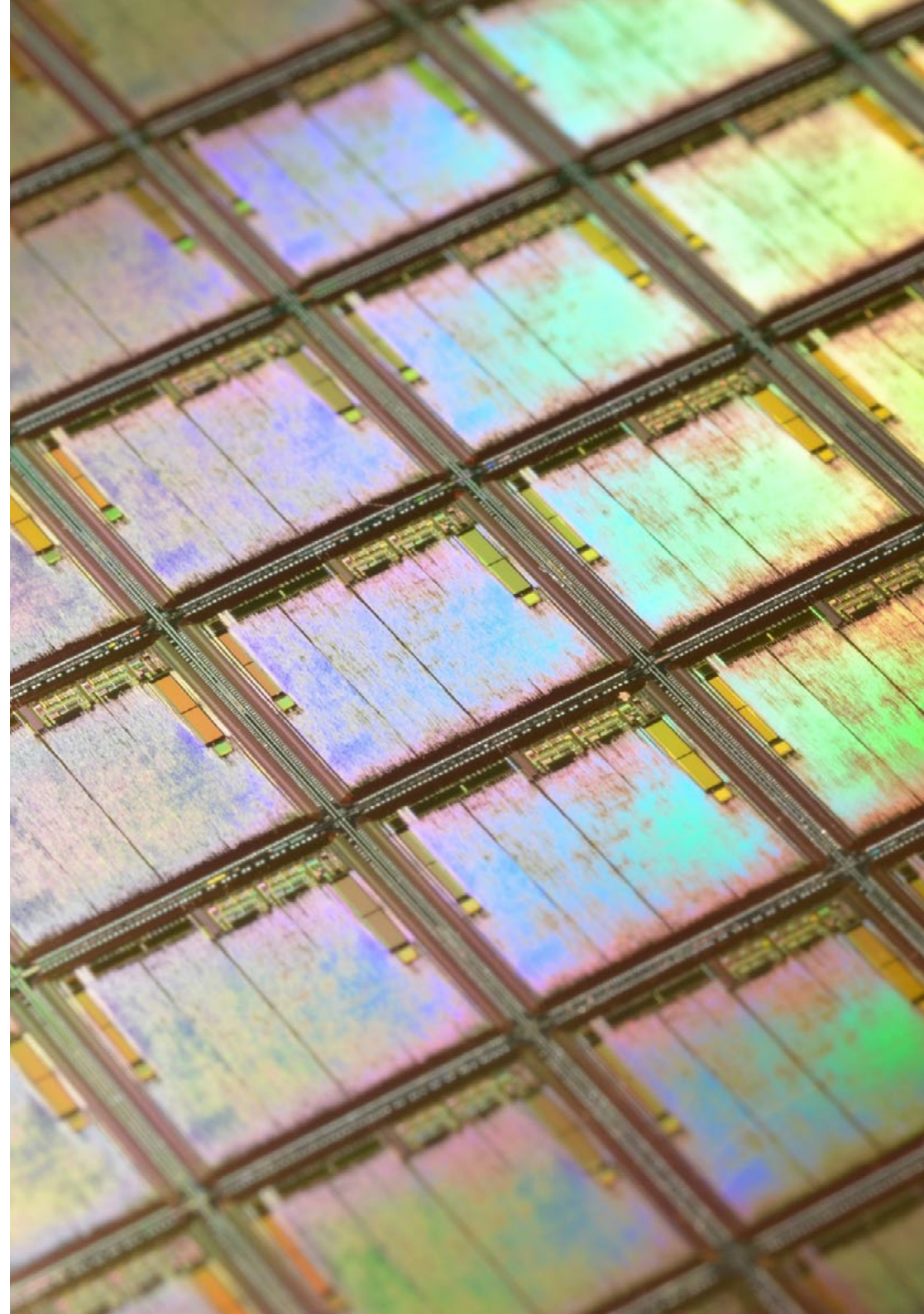
## Allgemeine Kompetenzen

---

- ◆ Entwickeln von Fachwissen über die verschiedenen Ebenen der Parallelisierung
- ◆ Analysieren einer Parallelisierungsstrategie anhand von Leistungsmetriken
- ◆ Bestimmen der Hauptmerkmale des parallelen und verteilten Rechnens, bevor die Kommunikation und Koordination zwischen seinen Komponenten behandelt wird
- ◆ Demonstrieren, dass in dieser Art von Systemen Interprozesskommunikation, Remote-Aufrufe oder indirekte Kommunikation stattfinden können
- ◆ Bestimmen der Leistungseinbußen von parallelen Anwendungen
- ◆ Analysieren von fortgeschrittenen Techniken zur Optimierung von parallelem Code, zur Optimierung der Kommunikation in Systemen mit verteiltem Speicher, zur Affinitätskontrolle, zum Lastausgleich und zur Verwaltung von parallelen Inputs/Outputs
- ◆ Untersuchen von hybriden Programmiermodellen für Systeme mit mehreren Hardwarebeschleunigern und hybriden Programmiermodellen für Systeme mit gemeinsamem/verteilttem Speicher



*Wenn Sie diese Kompetenzen zu Ihren besonderen Fähigkeiten hinzufügen, werden Sie ein viel attraktiverer Kandidat für die wichtigsten Positionen als IT-Entwickler"*







## Spezifische Kompetenzen

---

- ◆ Bestimmen der Probleme der Skalierbarkeit und Leistung, die durch die Zerlegung paralleler Prozesse gelöst werden können
- ◆ Analysieren der Merkmale eines Vorschlags für *Shared-Memory*-Parallelität, *Message-Passing*-Parallelität und Parallelität auf GPUs sowie des Hybrid-Szenarios
- ◆ Ermitteln des Bedarfs an Prozessausfallsicherheit und des Stakeholder-Modells bei der Lösung aktueller Computerprobleme
- ◆ Aufzeigen von Beispielen oder Fällen, in denen die parallele Zerlegung eine erfolgreiche Anwendung in Bezug auf Leistung oder Skalierbarkeit hatte
- ◆ Analysieren und Entwerfen paralleler Algorithmen
- ◆ Kompilieren der wichtigsten Primitive von MPI, OpenMP, OpenCL/CUDA
- ◆ Untersuchen von Vektor- und Matrixprozessen
- ◆ Analysieren der parallelen und verteilten Programmierung, Programmiersprachen, Tools und Entwurfsmuster
- ◆ Identifizieren der Elemente, die die Verbindung von verteilten Netzwerken ermöglichen
- ◆ Begründen der Entwurfsschritte für verteilte Systeme
- ◆ Evaluieren der verschiedenen Arten der Datenreplikation in bestehenden Systemen
- ◆ Zusammenstellen von anwendbaren praktischen Sicherheitsansätzen



# 04

# Kursleitung

In ihrem unermüdlichen Streben nach höchster Qualität in der Lehre hat TECH in diesem Programm eine Gruppe von Informatikern mit internationaler Erfahrung in einer Vielzahl von Projekten im Bereich der parallelen und verteilten Datenverarbeitung zusammengeführt. Auf diese Weise erwerben die Studenten Kenntnisse, die nicht nur auf ihre berufliche Entwicklung ausgerichtet sind, sondern auch aus der eigenen Erfahrung des Lehrkörpers in der Leitung multidisziplinärer IT-Teams stammen.





“

*Entwickeln Sie Ihr akademisches  
und berufliches Potenzial mit den  
besten Dozenten, die Sie während  
Ihres gesamten Studiums begleiten”*

## Leitung



### Hr. Olalla Bonal, Martín

- ♦ Senior Manager der Blockchain-Praxis bei EY
- ♦ Technischer Spezialist für Blockchain-Kunden bei IBM
- ♦ Direktor für Architektur bei Blocknitive
- ♦ Teamkoordinator für nicht relationale verteilte Datenbanken bei wedoIT (Tochtergesellschaft von IBM)
- ♦ Infrastruktur-Architekt bei Bankia
- ♦ Leiter der Layout-Abteilung bei T-Systems
- ♦ Abteilungskoordinator für Bing Data España SL

## Professoren

### Hr. Villot Guisán, Pablo

- ♦ Chief Information Officer, Chief Technical Officer und Gründer von New Tech & Talent
- ♦ Technologieexperte bei KPMG Spanien
- ♦ Blockchain Architekt bei Everis
- ♦ J2EE-Entwickler für den Bereich Handelslogistik bei Inditex
- ♦ Hochschulabschluss in Computertechnik an der Universität von La Coruña
- ♦ Microsoft MSCA-Zertifizierung: *Cloud Platform*

### Dr. Blanco, Eduardo

- ♦ Spezialist für Computerwissenschaften
- ♦ Dozent an der Universität Simón Bolívar
- ♦ Promotion in Informatik an der Universität Simón Bolívar
- ♦ Ingenieur in Computertechnik von der Universität Simón Bolívar
- ♦ Masterstudiengang in Computerwissenschaften, Universität Simón Bolívar



**Dr. Almendras Aruzamen, Luis Fernando**

- ◆ Ingenieur für Daten und Business Intelligence, Solutio Gruppe, Madrid
- ◆ Dateningenieur bei Indizen
- ◆ Daten- und Business Intelligence-Ingenieur bei Tecnología y Personas
- ◆ Ingenieur für Datenbank-, *Big Data*- und *Business Intelligence*-Support bei Equinix
- ◆ Daten-Ingenieur, Jalasoft
- ◆ Produktmanager und verantwortlich für den Bereich Business Analytics bei Goja
- ◆ Stellvertretender Business Intelligence Manager, VIVA Nuevatel PC's
- ◆ Verantwortlich für den Bereich *Datawarehouse* und *Big Data* bei Viva
- ◆ Leiter der Softwareentwicklung bei Intersoft
- ◆ Hochschulabschluss in Informatik an der Höheren Universität von San Simón
- ◆ Promotion in Computertechnik, Universität Complutense von Madrid
- ◆ Masterstudiengang in Computertechnik an der Universität Complutense von Madrid
- ◆ Masterstudiengang in Informationssysteme und Technologiemanagement an der Höheren Universität von San Simón
- ◆ Internationaler Ausbilder: Oracle Database, Proydesa-Oracle, Argentinien
- ◆ Zertifizierung als Project Management Professional, Beratungsunternehmen Alcances, Chile

**Dr. Carratalá Sáez, Rocío**

- ◆ Forscherin mit Spezialisierung auf Computerwissenschaften
- ◆ Dozentin für Universitätsstudien im Bereich Informatik
- ◆ Promotion in Informatik an der Universität Jaume I
- ◆ Hochschulabschluss in Computermathematik von der Universität Jaume I
- ◆ Masterstudiengang in Paralleler und Verteilter Datenverarbeitung an der Polytechnischen Universität von Valencia
- ◆ Spezialisierungskurse in den Bereichen Informatik, Mathematik und Werkzeuge für die akademische Forschung

**Hr. Gozalo Fernández, Juan Luis**

- ◆ Blockchain-basierter Produktmanager für Open Canarias
- ◆ Blockchain DevOps Manager bei Alastria
- ◆ Direktor für Service Level-Technologie bei Santander Spanien
- ◆ Manager für die Entwicklung der mobilen Anwendung Tinkerlink bei Cronos Telecom
- ◆ Technischer Direktor für IT-Service-Management bei Barclays Bank Spanien
- ◆ Hochschulabschluss in Computertechnik an der UNED
- ◆ Spezialisierung auf *Deep Learning* bei DeepLearning.ai

**Hr. Gómez Gómez, Borja**

- ◆ Leiter der Geschäftsentwicklung für Cloud Innovation bei Oracle
- ◆ Leiter für Blockchain und Architekturösungen vor dem Verkauf bei Paradigma Digital
- ◆ Senior IT-Architekt und Berater bei Atmira
- ◆ SOA-Architekt und -Berater bei TCP SI
- ◆ Analyst und Berater bei Everis
- ◆ Hochschulabschluss in Computertechnik an der Universität Complutense von Madrid
- ◆ Masterstudiengang in Science Computer Engineering an der Universität Complutense von Madrid

# 05 Struktur und Inhalt

Um ein besseres Studium und einen besseren Kompetenzerwerb zu ermöglichen, hat TECH die effektivsten Lehrmethoden in dieses Universitätsprogramm integriert. Dank des *Relearnings* können die Studenten die Zeit, die sie für den Erwerb der wichtigsten Kenntnisse des Programms aufwenden müssen, erheblich verkürzen. Dies wird auch durch die große Menge an audiovisuellem Material, ergänzender Lektüre und praktischen Übungen unterstützt, die zur Festigung des gesamten Lehrstoffs beitragen.



“

*Praktische Übungen anhand von Fallbeispielen und Videos, die von den Professoren selbst detailliert ausgearbeitet wurden, sind der Schlüssel zu Ihrem Erfolg in diesem Universitätsprogramm"*



## Modul 1. Parallelität in der parallelen und verteilten Datenverarbeitung

- 1.1. Parallele Verarbeitung:
  - 1.1.1. Parallele Verarbeitung:
  - 1.1.2. Parallele Verarbeitung im Computerwesen. Zweck
  - 1.1.3. Parallele Verarbeitung: Analyse
- 1.2. Parallele Systeme
  - 1.2.1. Das parallele System
  - 1.2.2. Ebenen der Parallelität
  - 1.2.3. Parallele Systemzusammensetzung
- 1.3. Prozessor-Architekturen
  - 1.3.1. Prozessorkomplexität
  - 1.3.2. Prozessor-Architektur. Arbeitsweise
  - 1.3.3. Prozessor-Architektur. Speicherorganisation
- 1.4. Netzwerke in der Parallelverarbeitung
  - 1.4.1. Arbeitsweise
  - 1.4.2. Kontrollstrategie
  - 1.4.3. Umschalttechniken
  - 1.4.4. Topologie
- 1.5. Parallele Architekturen
  - 1.5.1. Algorithmen
  - 1.5.2. Kopplung
  - 1.5.3. Kommunikation
- 1.6. Leistung der parallelen Datenverarbeitung
  - 1.6.1. Entwicklung der Leistung
  - 1.6.2. Performance-Messungen
  - 1.6.3. Parallele Datenverarbeitung. Fallstudien
- 1.7. Flynn-Taxonomie
  - 1.7.1. MIMD: Gemeinsamer Speicher
  - 1.7.2. MIMD: Verteilter Speicher
  - 1.7.3. MIMD: Hybride Systeme
  - 1.7.4. Datenfluss

- 1.8. Formen der Parallelität: TLP (*Thread Level Parallelism*)
  - 1.8.1. Formen der Parallelität: TLP (*Thread Level Parallelism*)
  - 1.8.2. *Coarse Grain*
  - 1.8.3. *Fine Grain*
  - 1.8.4. SMT
- 1.9. Formen der Parallelität: DLP (*Data Level Parallelism*)
  - 1.9.1. Formen der Parallelität: DLP (*Data Level Parallelism*)
  - 1.9.2. *Short Vector Processing*
  - 1.9.3. *Vector Processors*
- 1.10. Formen der Parallelität: ILP (*Instruction Level Parallelism*)
  - 1.10.1. Formen der Parallelität: ILP (*Instruction Level Parallelism*)
  - 1.10.2. Segmentierter Prozessor
  - 1.10.3. Superskalarer Prozessor
  - 1.10.4. *Very Long Instruction Word* Prozessor (VLIW)

## Modul 2. Parallele Zerlegung in der parallelen und verteilten Datenverarbeitung

- 2.1. Parallele Zerlegung in der parallelen und verteilten Datenverarbeitung
  - 2.1.1. Parallele Verarbeitung:
  - 2.1.2. Architekturen
  - 2.1.3. Supercomputer
- 2.2. Parallele Hardware und parallele Software
  - 2.2.1. Serielle Systeme
  - 2.2.2. Parallele Hardware
  - 2.2.3. Parallele Software
  - 2.2.4. Eingabe und Ausgabe
  - 2.2.5. Leistung
- 2.3. Parallele Skalierbarkeit und wiederkehrende Leistungsprobleme
  - 2.3.1. Parallelität
  - 2.3.2. Parallele Skalierbarkeit
  - 2.3.3. Wiederkehrende Leistungsprobleme
- 2.4. Parallelität bei gemeinsamem Speicher
  - 2.4.1. Parallelität bei gemeinsamem Speicher
  - 2.4.2. OpenMP und Pthreads
  - 2.4.3. Parallelität bei gemeinsamem Speicher. Beispiele



- 2.5. Grafikverarbeitungseinheit (GPU)
  - 2.5.1. Grafikverarbeitungseinheit (GPU)
  - 2.5.2. Vereinheitlichte Rechengeratearchitektur (CUDA)
  - 2.5.3. Vereinheitlichte Rechengeratearchitektur. Beispiele
- 2.6. Message Passing-Systeme
  - 2.6.1. Message Passing-Systeme
  - 2.6.2. MPI. Message Passing-Schnittstelle
  - 2.6.3. Nachrichtenübermittlungssysteme. Beispiele
- 2.7. Hybride Parallelisierung mit MPI und OpenMP
  - 2.7.1. Hybride Programmierung
  - 2.7.2. MPI/OpenMP-Programmiermodelle
  - 2.7.3. Hybride Zerlegung und Mapping
- 2.8. MapReduce-Berechnungen
  - 2.8.1. Hadoop
  - 2.8.2. Andere Datenverarbeitungssysteme
  - 2.8.3. Parallele Datenverarbeitung. Beispiele
- 2.9. Aktorenmodell und reaktive Prozesse
  - 2.9.1. Aktorenmodell
  - 2.9.2. Reaktive Prozesse
  - 2.9.3. Aktoren und reaktive Prozesse. Beispiele
- 2.10. Szenarien für parallele Datenverarbeitung
  - 2.10.1. Audio- und Bildverarbeitung
  - 2.10.2. Statistik/Data Mining
  - 2.10.3. Paralleles Sortieren
  - 2.10.4. Parallele Matrixoperationen

### Modul 3. Kommunikation und Koordination in Computersystemen

- 3.1. Parallele und verteilte Datenverarbeitungsprozesse
  - 3.1.1. Parallele und verteilte Datenverarbeitungsprozesse
  - 3.1.2. Prozesse und Threads
  - 3.1.3. Virtualisierung
  - 3.1.4. Clients und Server
- 3.2. Kommunikation bei paralleler Datenverarbeitung
  - 3.2.1. Kommunikation bei paralleler Datenverarbeitung
  - 3.2.2. Mehrschichtige Protokolle
  - 3.2.3. Kommunikation bei paralleler Datenverarbeitung. Typologie
- 3.3. *Remote Procedure Call*
  - 3.3.1. Wie RPC (*Remote Procedure Call*) funktioniert
  - 3.3.2. Parameterübergabe
  - 3.3.3. Asynchroner RPC
  - 3.3.4. Remote-Prozedur. Beispiele
- 3.4. Nachrichtenorientierte Kommunikation
  - 3.4.1. Transiente nachrichtenorientierte Kommunikation
  - 3.4.2. Persistente nachrichtenorientierte Kommunikation
  - 3.4.3. Nachrichtenorientierte Kommunikation. Beispiele
- 3.5. Flussorientierte Kommunikation
  - 3.5.1. Unterstützung für kontinuierliche Medien
  - 3.5.2. Datenfluss und Dienstqualität
  - 3.5.3. Stream-Synchronisierung
  - 3.5.4. Flussorientierte Kommunikation. Beispiele
- 3.6. Multicast-Kommunikation
  - 3.6.1. Multicast auf Anwendungsebene
  - 3.6.2. Codebasierte Datenverbreitung
  - 3.6.3. Multicast-Kommunikation. Beispiele
- 3.7. Andere Arten der Kommunikation
  - 3.7.1. Remote-Methodenaufruf
  - 3.7.2. Webdienste / SOA / REST
  - 3.7.3. Event-Benachrichtigung
  - 3.7.4. Mobile Agenten

- 3.8. Namensdienst
  - 3.8.1. Computer-Namensdienste
  - 3.8.2. Benennungsdienste und *Domain Name System*
  - 3.8.3. Verzeichnisdienste
- 3.9. Synchronisierung
  - 3.9.1. Synchronisierung der Uhr
  - 3.9.2. Logische Uhren, gegenseitiger Ausschluss und globale Positionierung von Knoten
  - 3.9.3. Auswahl der Algorithmen
- 3.10. Kommunikation. Koordinierung und Einigung
  - 3.10.1. Koordinierung und Einigung
  - 3.10.2. Koordinierung und Einigung. Konsens und Probleme
  - 3.10.3. Kommunikation und Koordination. Aktualität

## Modul 4. Analyse und Programmierung von parallelen Algorithmen

- 4.1. Parallele Algorithmen
  - 4.1.1. Problem-Zerlegung
  - 4.1.2. Daten-Abhängigkeiten
  - 4.1.3. Implizite und explizite Parallelität
- 4.2. Parallele Programmierparadigmen
  - 4.2.1. Parallele Programmierung mit gemeinsamem Speicher
  - 4.2.2. Parallele Programmierung mit verteiltem Speicher
  - 4.2.3. Hybride Parallelprogrammierung
  - 4.2.4. Heterogenes Rechnen - CPU + GPU
  - 4.2.5. Quantencomputing. Neue Programmiermodelle mit impliziter Parallelität
- 4.3. Parallele Programmierung mit gemeinsamem Speicher
  - 4.3.1. Parallele Programmiermodelle mit gemeinsamem Speicher (*Shared-Memory*)
  - 4.3.2. Parallele *Shared-Memory*-Algorithmen
  - 4.3.3. Parallele *Shared-Memory*-Programmierbibliotheken
- 4.4. OpenMP
  - 4.4.1. OpenMP
  - 4.4.2. Ausführen und Debuggen von Programmen mit OpenMP
  - 4.4.3. Parallele Algorithmen mit gemeinsamem Speicher in OpenMP
- 4.5. Parallele *Message Passing*-Programmierung
  - 4.5.1. *Primitive Message Passing*
  - 4.5.2. Kommunikation und kollektive Datenverarbeitung
  - 4.5.3. Parallele Algorithmen zum *Message Passing*
  - 4.5.4. Bibliotheken für die parallele Programmierung mit *Message Passing*
- 4.6. *Message Passing Interface* (MPI)
  - 4.6.1. *Message Passing Interface* (MPI)
  - 4.6.2. Ausführen und Debuggen von Programmen mit MPI
  - 4.6.3. Parallele Algorithmen zum *Message Passing* mit MPI
- 4.7. Hybride Parallelprogrammierung
  - 4.7.1. Hybride Parallelprogrammierung
  - 4.7.2. Ausführen und Debuggen von hybriden Programmen
  - 4.7.3. Hybride parallele MPI-OpenMP-Algorithmen
- 4.8. Parallele Programmierung mit heterogener Datenverarbeitung
  - 4.8.1. Parallele Programmierung mit heterogener Datenverarbeitung
  - 4.8.2. CPU vs. GPU
  - 4.8.3. Parallele Algorithmen mit heterogener Datenverarbeitung
- 4.9. OpenCL und CUDA
  - 4.9.1. OpenCL vs. CUDA
  - 4.9.2. Ausführung und Fehlersuche bei parallelen Programmen mit heterogener Datenverarbeitung
  - 4.9.3. Parallele Algorithmen mit heterogener Datenverarbeitung
- 4.10. Entwurf von parallelen Algorithmen
  - 4.10.1. Entwurf von parallelen Algorithmen
  - 4.10.2. Problem und Kontext
  - 4.10.3. Automatische Parallelisierung vs. Manuelle Parallelisierung
  - 4.10.4. Partitionierung des Problems
  - 4.10.5. Kommunikation in der Datenverarbeitung

## Modul 5. Parallele Architekturen

- 5.1. Parallele Architekturen
  - 5.1.1. Parallele Systeme. Klassifizierung
  - 5.1.2. Quellen der Parallelität
  - 5.1.3. Parallelität und Prozessoren
- 5.2. Leistung von parallelen Systemen
  - 5.2.1. Leistungsmetriken und -messungen
  - 5.2.2. *Speed-up*
  - 5.2.3. Granularität von parallelen Systemen
- 5.3. Vektorprozessoren
  - 5.3.1. Basis-Vektorprozessor
  - 5.3.2. Verschachtelter Speicher
  - 5.3.3. Leistung des Vektorprozessors
- 5.4. Matrix-Prozessoren
  - 5.4.1. Grundlegende Organisation
  - 5.4.2. Programmierung in Matrixprozessoren
  - 5.4.3. Programmierung in Matrixprozessoren. Praktisches Beispiel
- 5.5. Verbindungsnetzwerke
  - 5.5.1. Verbindungsnetzwerke
  - 5.5.2. Topologie, Flusskontrolle und Routing
  - 5.5.3. Verbindungsnetzwerke. Klassifizierung nach der Topologie
- 5.6. Multiprozessoren
  - 5.6.1. Multiprozessor-Verbindungsnetzwerke
  - 5.6.2. Speicher und Cache-Konsistenz
  - 5.6.3. Abfrage-Protokolle
- 5.7. Synchronisierung
  - 5.7.1. Sperren (gegenseitiger Ausschluss)
  - 5.7.2. P2P-Synchronisierungsereignisse
  - 5.7.3. Globale Synchronisierungsereignisse
- 5.8. Multicomputer
  - 5.8.1. Multicomputer-Verbindungsnetzwerke
  - 5.8.2. Vermittlungsschicht
  - 5.8.3. Routing-Schicht

- 5.9. Erweiterte Architekturen
  - 5.9.1. Datenfluss-Maschinen
  - 5.9.2. Andere Architekturen
- 5.10. Parallele und verteilte Programmierung
  - 5.10.1. Parallele Programmiersprachen
  - 5.10.2. Parallele Programmierwerkzeuge
  - 5.10.3. Entwurfsmuster
  - 5.10.4. Gleichzeitigkeit von parallelen und verteilten Programmiersprachen

## Modul 6. Parallele Leistung

- 6.1. Leistung von parallelen Algorithmen
  - 6.1.1. Das Ahmdalsche Gesetz
  - 6.1.2. Gustarsons Gesetz
  - 6.1.3. Leistungsmetriken und Skalierbarkeit von parallelen Algorithmen
- 6.2. Vergleich von parallelen Algorithmen
  - 6.2.1. *Benchmarking*
  - 6.2.2. Mathematische Analyse von parallelen Algorithmen
  - 6.2.3. Asymptotische Analyse von parallelen Algorithmen
- 6.3. Hardware-Ressourcen-Beschränkungen
  - 6.3.1. Speicher
  - 6.3.2. Verarbeitung
  - 6.3.3. Kommunikation
  - 6.3.4. Dynamische Ressourcenpartitionierung
- 6.4. Leistung von parallelen Programmen mit gemeinsamem Speicher
  - 6.4.1. Optimale Aufgabenpartitionierung
  - 6.4.2. *Thread-Affinität*
  - 6.4.3. SIMD-Parallelität
  - 6.4.4. Parallele Programme mit gemeinsamem Speicher. Beispiele
- 6.5. Leistung von parallelen Programmen durch *Message Passing*
  - 6.5.1. Leistung von parallelen Programmen durch *Message Passing*
  - 6.5.2. Optimierung der Kommunikation in MPI
  - 6.5.3. Affinitätskontrolle und Lastausgleich
  - 6.5.4. Paralleler I/O
  - 6.5.5. Parallele *Message-Passing*-Programme. Beispiele



- 6.6. Hybride parallele Programmleistung
  - 6.6.1. Hybride parallele Programmleistung
  - 6.6.2. Hybride Programmierung für Systeme mit gemeinsamem/verteilter Speicher
  - 6.6.3. Hybride parallele Programme. Beispiele
- 6.7. Leistung von heterogenen Datenverarbeitungsprogrammen
  - 6.7.1. Leistung von heterogenen Datenverarbeitungsprogrammen
  - 6.7.2. Hybride Programmierung für Systeme mit mehreren Hardware-Beschleunigern
  - 6.7.3. Programme mit heterogener Datenverarbeitung. Beispiele
- 6.8. Leistungsanalyse von parallelen Algorithmen
  - 6.8.1. Leistungsanalyse von parallelen Algorithmen
  - 6.8.2. Leistungsanalyse von parallelen Algorithmen. Tools
  - 6.8.3. Leistungsanalyse von parallelen Algorithmen. Empfehlungen
- 6.9. Parallele Muster
  - 6.9.1. Parallele Muster
  - 6.9.2. Wichtigste parallele Muster
  - 6.9.3. Parallele Muster. Vergleich
- 6.10. Leistungsstarke parallele Programme
  - 6.10.1. Prozesse
  - 6.10.2. Leistungsstarke parallele Programme
  - 6.10.3. Leistungsstarke parallele Programme. Reale Anwendungen

## Modul 7. Verteilte Systeme in der Datenverarbeitung

- 7.1. Verteilte Systeme
  - 7.1.1. Verteilte Systeme (DS)
  - 7.1.2. Beweis des CAP-Theorems (oder Brewer's Conjecture)
  - 7.1.3. Programmierirrtümer über verteilte Systeme
  - 7.1.4. Ubiquitäre Datenverarbeitung
- 7.2. Verteilte Systeme. Eigenschaften
  - 7.2.1. Heterogenität
  - 7.2.2. Erweiterbarkeit
  - 7.2.3. Sicherheit
  - 7.2.4. Skalierbarkeit
  - 7.2.5. Fehlertoleranz
  - 7.2.6. Parallelität
  - 7.2.7. Transparenz
- 7.3. Vernetzung und Zusammenschaltung von verteilten Netzwerken
  - 7.3.1. Netzwerke und verteilte Systeme. Leistung von Netzwerken
  - 7.3.2. Verfügbare Netzwerke zur Erstellung eines verteilten Systems. Typologie
  - 7.3.3. Verteilte vs. zentralisierte Netzwerkprotokolle
  - 7.3.4. Zusammenschaltung von Netzwerken. Internet
- 7.4. Kommunikation zwischen verteilten Prozessen
  - 7.4.1. Kommunikation zwischen Knoten eines DS. Probleme und Ausfälle
  - 7.4.2. Mechanismen, die zusätzlich zu RPC und RDMA implementiert werden können, um Ausfälle zu vermeiden
  - 7.4.3. Mechanismen, die in Software implementiert werden können, um Ausfälle zu vermeiden
- 7.5. Entwurf verteilter Systeme
  - 7.5.1. Effizienter Entwurf von verteilten Systemen (DS)
  - 7.5.2. Muster für die Programmierung verteilter Systeme (DS)
  - 7.5.3. Service-orientierte Architektur (*Service Oriented Architecture (SOA)*)
  - 7.5.4. *Service Orchestration* und *Microservices Data Management*
- 7.6. Betrieb von verteilten Systemen
  - 7.6.1. Überwachung von Systemen
  - 7.6.2. Implementierung eines effizienten Protokollierungssystems (*Logging*) in einem DS
  - 7.6.3. Überwachung in verteilten Netzwerken
  - 7.6.4. Verwendung eines Überwachungstools für ein SD Prometheus und Grafana

- 7.7. Systemreplikation
  - 7.7.1. Systemreplikation. Typologien
  - 7.7.2. Unveränderliche Architekturen
  - 7.7.3. Container-Systeme und Virtualisierung von Systemen als verteilte Systeme
  - 7.7.4. *Blockchain*-Netzwerke als verteilte Systeme
- 7.8. Verteilte Multimedia-Systeme
  - 7.8.1. Verteilter Bild- und Videoaustausch. Problemstellung
  - 7.8.2. Multimedia-Objekt-Server
  - 7.8.3. Netzwerktopologie für ein Multimedia-System
  - 7.8.4. Analyse von verteilten Multimedia-Systemen: Netflix, Amazon, Spotify etc.
  - 7.8.5. Verteilte Multimedia-Systeme im Bildungswesen
- 7.9. Verteilte Dateisysteme
  - 7.9.1. Verteilter Dateiaustausch. Problemstellung
  - 7.9.2. Anwendbarkeit des CAP-Theorems auf Datenbanken
  - 7.9.3. Verteilte Web-Dateisysteme: Akamai
  - 7.9.4. Verteilte Dokumentdateisysteme IPFS
  - 7.9.5. Verteilte Datenbanksysteme
- 7.10. Sicherheitsansätze für verteilte Systeme
  - 7.10.1. Sicherheit verteilter Systeme
  - 7.10.2. Bekannte Angriffe auf verteilte Systeme
  - 7.10.3. Tools zum Testen der Sicherheit eines DS

## Modul 8. Parallele Datenverarbeitung angewandt auf Cloud-Umgebungen

- 8.1. *Cloud Computing*
  - 8.1.1. Aktueller Stand der IT-Landschaft
  - 8.1.2. Die Cloud
  - 8.1.3. *Cloud Computing*
- 8.2. Sicherheit und Ausfallsicherheit in der Cloud
  - 8.2.1. Regionen, Verfügbarkeit und Fehlerzonen
  - 8.2.2. Verwaltung von *Cloud-Tenant* oder -Konten
  - 8.2.3. Identitäts- und Zugriffskontrolle in der Cloud
- 8.3. *Cloud-Networking*
  - 8.3.1. Softwaredefinierte virtuelle Netzwerke
  - 8.3.2. Netzwerkkomponenten von *Software Defined Networking*
  - 8.3.3. Verbindung zu anderen Systemen
- 8.4. Cloud-Dienste
  - 8.4.1. Infrastruktur als Dienstleistung
  - 8.4.2. Plattform als Dienst
  - 8.4.3. *Serverless*-Rechnen
  - 8.4.4. Software als Dienstleistung
- 8.5. Cloud-Speicher
  - 8.5.1. Cloud-Blockspeicher
  - 8.5.2. Speicherung von Dateien in der Cloud
  - 8.5.3. Speicherung von Objekten in der Cloud
- 8.6. Cloud-Interaktion und Überwachung
  - 8.6.1. Überwachung und Verwaltung der Cloud
  - 8.6.2. Interaktion mit der Cloud: Verwaltungskonsole
  - 8.6.3. Interaktion mit der *Command Line Interface*
  - 8.6.4. Interaktion auf Basis von APIs
- 8.7. *Cloud-Native* Entwicklung
  - 8.7.1. *Cloud-Native* Entwicklung
  - 8.7.2. Container und Container-Orchestrierungsplattformen
  - 8.7.3. Kontinuierliche Cloud-Integration
  - 8.7.4. Nutzung von Cloud-Ereignissen

- 8.8. Infrastruktur als Code in der Cloud
  - 8.8.1. Automatisierung der Verwaltung und Bereitstellung in der Cloud
  - 8.8.2. Terraform
  - 8.8.3. Integration mit *Scripting*
- 8.9. Erstellen einer hybriden Infrastruktur
  - 8.9.1. Zusammenschaltung
  - 8.9.2. Zusammenschaltung mit dem *Datacenter*
  - 8.9.3. Zusammenschaltung mit anderen Clouds
- 8.10. Hochleistungs-Computing
  - 8.10.1. Hochleistungs-Computing
  - 8.10.2. Erstellung eines Hochleistungsclusters
  - 8.10.3. Anwendung von Hochleistungs-Computing

## Modul 9. Modelle und formale Semantik. Programmierung für verteilte Datenverarbeitung

- 9.1. Logisches Datenmodell
  - 9.1.1. Semantische Datenmodelle
  - 9.1.2. Semantische Datenmodelle. Verwendungszwecke
  - 9.1.3. Semantische Datenmodelle. Anwendungen
- 9.2. Semantisches Modell von Programmiersprachen
  - 9.2.1. Sprachverarbeitung
  - 9.2.2. Übersetzen und Interpretieren
  - 9.2.3. Hybride Sprachen
- 9.3. Modelle der Datenverarbeitung
  - 9.3.1. Monolithische Datenverarbeitung
  - 9.3.2. Parallele Datenverarbeitung
  - 9.3.3. Verteilte Datenverarbeitung
  - 9.3.4. Kooperative Datenverarbeitung (P2P)
- 9.4. Parallele Datenverarbeitung
  - 9.4.1. Parallele Architektur
  - 9.4.2. Hardware
  - 9.4.3. Software

- 9.5. Verteiltes Modell. *Grid Computing*
  - 9.5.1. *Grid-Computing*-Architektur
  - 9.5.2. *Grid-Computing*-Architektur. Analyse
  - 9.5.3. *Grid-Computing*-Architektur. Anwendungen
- 9.6. Verteiltes Modell. *Cluster Computing*
  - 9.6.1. *Cluster Computing*-Architektur
  - 9.6.2. *Cluster Computing*-Architektur. Analyse
  - 9.6.3. *Cluster Computing*-Architektur. Anwendungen
- 9.7. *Cluster Computing*. Aktuelle Tools zur Implementierung. Hypervisoren
  - 9.7.1. Marktkonkurrenten
  - 9.7.2. VMware Hypervisor
  - 9.7.3. Hyper-V
- 9.8. Verteiltes Modell. *Cloud Computing*
  - 9.8.1. Architektur des *Cloud Computing*
  - 9.8.2. Architektur des *Cloud Computing*. Analyse
  - 9.8.3. Architektur des *Cloud Computing*. Anwendungen
- 9.9. Verteiltes Modell. *Cloud Computing* Amazon
  - 9.9.1. *Cloud Computing* Amazon. Funktionalitäten
  - 9.9.2. *Cloud Computing* Amazon. Lizenzierung
  - 9.9.3. *Cloud Computing* Amazon. Referenzarchitektur
- 9.10. Verteiltes Modell. *Cloud Computing* Microsoft
  - 9.10.1. *Cloud Computing* Microsoft. Funktionalitäten
  - 9.10.2. *Cloud Computing* Microsoft. Lizenzierung
  - 9.10.3. *Cloud Computing* Microsoft. Referenzarchitektur



**Modul 10.** Anwendungen der parallelen und verteilten Datenverarbeitung

- 10.1. Parallele und verteilte Datenverarbeitung in heutigen Anwendungen
  - 10.1.1. Hardware
  - 10.1.2. Software
  - 10.1.3. Die Bedeutung des Timings
- 10.2. Das Klima. Der Klimawandel
  - 10.2.1. Klimaanwendungen. Datenquellen
  - 10.2.2. Klimaanwendungen. Datenmengen
  - 10.2.3. Klimaanwendungen. Real-Time
- 10.3. GPU Parallele Datenverarbeitung
  - 10.3.1. GPU Parallele Datenverarbeitung
  - 10.3.2. GPUs vs. CPU. GPU-Nutzung
  - 10.3.3. GPU. Beispiele
- 10.4. *Smart Grid*. Datenverarbeitung in Stromnetzen
  - 10.4.1. *Smart Grid*
  - 10.4.2. Konzeptionelle Modelle. Beispiele
  - 10.4.3. *Smart Grid*. Beispiel
- 10.5. Verteilte Engine. *ElasticSearch*
  - 10.5.1. Verteilte Engine. *ElasticSearch*
  - 10.5.2. Architektur mit *ElasticSearch*. Beispiele
  - 10.5.3. Verteilte Engine. Anwendungsbeispiele
- 10.6. *Big Data Framework*
  - 10.6.1. *Big Data Framework*
  - 10.6.2. Architektur der fortgeschrittenen Tools
  - 10.6.3. *Big Data* in der verteilten Datenverarbeitung
- 10.7. *In-Memory*-Datenbank
  - 10.7.1. *In-Memory*-Datenbank
  - 10.7.2. Redis-Lösung. Fallstudie
  - 10.7.3. Einsatz von *In-Memory*-Datenbanklösungen
- 10.8. *Blockchain*
  - 10.8.1. *Blockchain*-Architektur. Komponenten
  - 10.8.2. Zusammenarbeit zwischen Knotenpunkten und Konsens
  - 10.8.3. *Blockchain*-Lösungen. Implementierungen

- 10.9. Verteilte Systeme in der Medizin
  - 10.9.1. Komponenten der Architektur
  - 10.9.2. Verteilte Systeme in der Medizin. Funktionsweise
  - 10.9.3. Verteilte Systeme in der Medizin. Anwendungen
- 10.10. Verteilte Systeme in der Luftfahrt
  - 10.10.1. Entwurf der Architektur
  - 10.10.2. Verteilte Systeme in der Luftfahrt. Funktionalitäten der Komponenten
  - 10.10.3. Verteilte Systeme in der Luftfahrt. Anwendungen



*Dieses Programm ist der Anstoß, den Sie brauchen, um den beruflichen Aufstieg zu erreichen, den Sie verdienen und für den Sie hart gearbeitet haben“*

# 06 Methodik

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: **das Relearning**. Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem **New England Journal of Medicine** als eines der effektivsten angesehen.





“

*Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen hinter sich lässt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"*

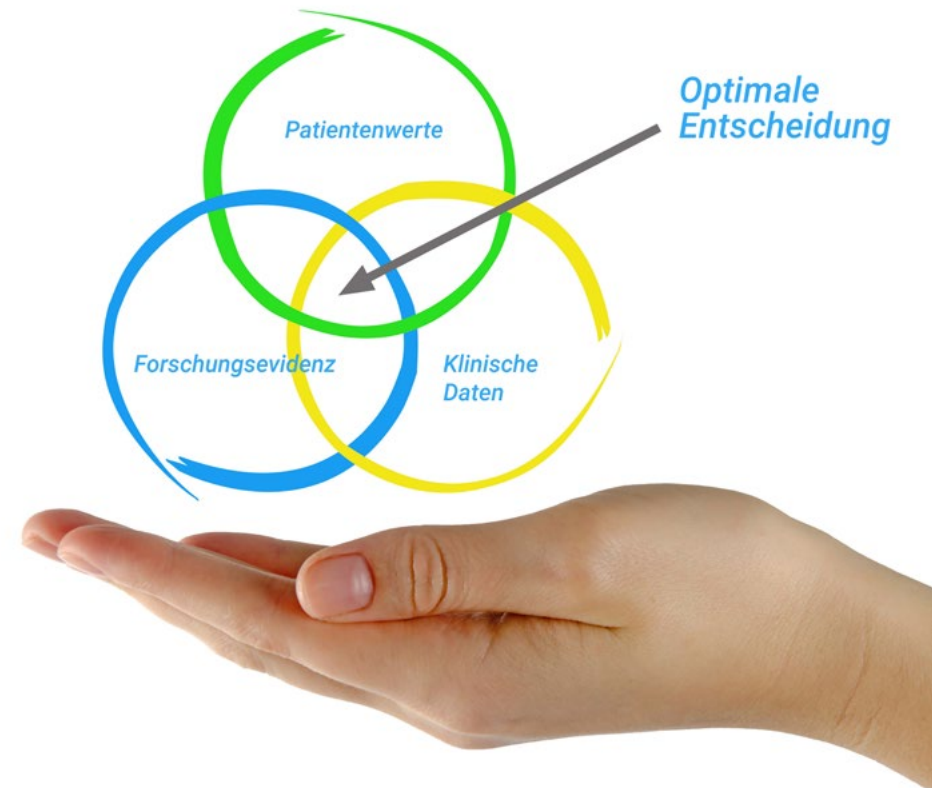


## Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

“

*Mit TECH werden Sie eine Art des Lernens erleben, die an den Grundlagen der traditionellen Universitäten auf der ganzen Welt rüttelt“*



*Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.*



*Der Student wird durch gemeinschaftliche Aktivitäten und reale Fälle lernen, wie man komplexe Situationen in realen Geschäftsumgebungen löst.*

## Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

**“** *Unser Programm bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein“*

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit Jurastudenten das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernen. Sie bestand darin, ihnen reale komplexe Situationen zu präsentieren, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen konnten, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode.

Während des gesamten Kurses werden die Studenten mit mehreren realen Fällen konfrontiert. Sie müssen ihr gesamtes Wissen integrieren, recherchieren, argumentieren und ihre Ideen und Entscheidungen verteidigen.

## Relearning Methodology

TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

*Im Jahr 2019 erzielten wir die besten  
Lernergebnisse aller spanischsprachigen  
Online-Universitäten der Welt.*

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft zu spezialisieren. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten spanischsprachigen Online-Universität zu verbessern.







In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -instrumente fortgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

*Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.*

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten Neurocognitive Context-Dependent E-Learning mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.

Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:



#### Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf das audiovisuelle Format angewendet, um die Online-Arbeitsmethode von TECH zu schaffen. All dies mit den neuesten Techniken, die in jedem einzelnen der Materialien, die dem Studenten zur Verfügung gestellt werden, qualitativ hochwertige Elemente bieten.



#### Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte Learning from an Expert festigt das Wissen und das Gedächtnis und schafft Vertrauen für zukünftige schwierige Entscheidungen.



#### Übungen für Fertigkeiten und Kompetenzen

Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.



#### Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u. a. In der virtuellen Bibliothek von TECH hat der Student Zugang zu allem, was er für seine Fortbildung benötigt.





#### Case Studies

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.



#### Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

Dieses einzigartige Bildungssystem für die Präsentation multimedialer Inhalte wurde von Microsoft als "Europäische Erfolgsgeschichte" ausgezeichnet.



#### Testing & Retesting

Die Kenntnisse des Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass der Student überprüfen kann, wie er seine Ziele erreicht.





07

# Qualifizierung

Der Privater Masterstudiengang in Parallele und Verteilte Datenverarbeitung garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Technologischen Universität ausgestellten Diplom.



“

*Schließen Sie dieses Programm erfolgreich ab und erhalten Sie Ihren Universitätsabschluss ohne lästige Reisen oder Formalitäten"*

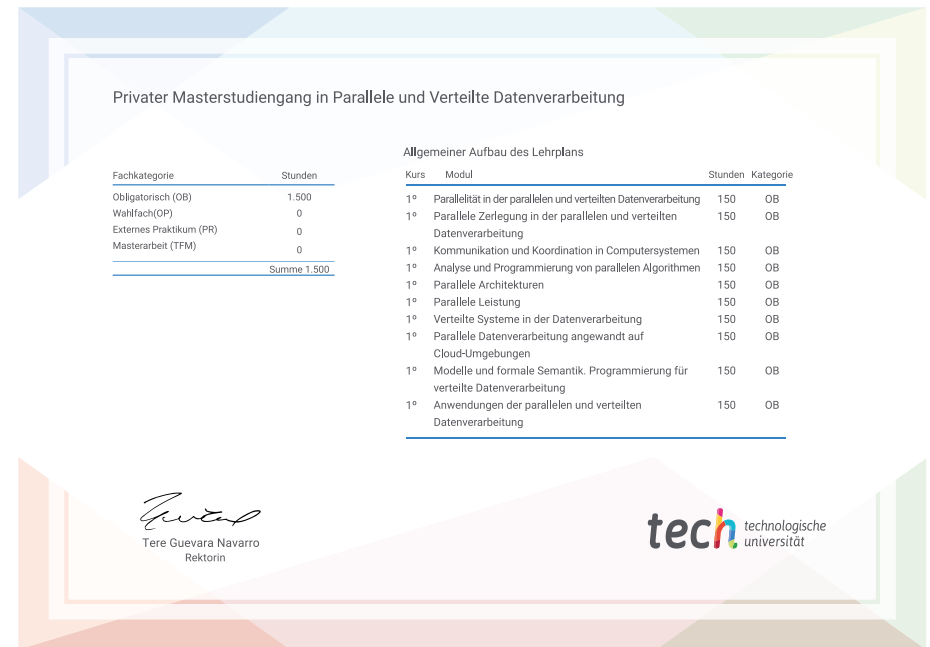
Dieser **Privater Masterstudiengang in Parallele und Verteilte Datenverarbeitung** enthält das vollständigste und aktuellste Programm auf dem Markt.

Sobald der Student die Prüfungen bestanden hat, erhält er/sie per Post\* mit Empfangsbestätigung das entsprechende Diplom, ausgestellt von der **TECH Technologischen Universität**.

Das von **TECH Technologische Universität** ausgestellte Diplom drückt die erworbene Qualifikation aus und entspricht den Anforderungen, die in der Regel von Stellenbörsen, Auswahlprüfungen und Berufsbildungsausschüssen verlangt werden.

Titel: **Privater Masterstudiengang in Parallele und Verteilte Datenverarbeitung**

Anzahl der offiziellen Arbeitsstunden: **1.500 Std.**



\*Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH EDUCATION die notwendigen Vorkehrungen treffen, um diese gegen eine zusätzliche Gebühr zu beschaffen.



zukunft

gesundheit vertrauen menschen  
erziehung information tutoren  
garantie akkreditierung unterricht  
institutionen technologie lernen  
gemeinschaft verpflichtung  
persönliche betreuung innovation  
wissen gegenwart qualität  
online-Ausbildung  
entwicklung institutionen  
virtuelles Klassenzimmer

**tech** technologische  
universität

**Privater Masterstudiengang**  
Parallele und Verteilte  
Datenverarbeitung

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Aufwand: 16 Std./Woche
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

# Privater Masterstudiengang

## Parallele und Verteilte Datenverarbeitung

