

Verteilte Systeme und Anwendungen

Code	VSA		
Fachbereich(e)	Enterprise Computing		
Studiengang /-gänge	BSc Informatik		
Vertiefungsrichtung(en)	-		
Art des Studiengangs	<input checked="" type="checkbox"/> Bachelor	<input type="checkbox"/> Master	<input type="checkbox"/> CAS/MAS/EMBA
Studienniveau *	<input type="checkbox"/> Basic	<input type="checkbox"/> Intermediate	<input checked="" type="checkbox"/> Advanced <input type="checkbox"/> Specialised
Typus **	<input type="checkbox"/> Core course	<input checked="" type="checkbox"/> Related course	<input type="checkbox"/> Minor course
ECTS-Credits	5		
Präsenzverpflichtung	100%		
Arbeitsaufwand in Std.	150		
Verantwortliche Ansprechperson	Fachbereichsleiter: Prof. Dr. Heinrich Zimmermann	Autor: Dr. Ilir Fetai	
Zu entwickelnde Kompetenzen	<ul style="list-style-type: none"> Die Studierenden kennen die grundlegenden Konzepte und Paradigmen verteilter Systeme und Anwendungen. Die Studierenden kennen die Ziele und Konzepte von Grid & Cloud Computing Die Studierende kennen die grundlegenden Herausforderungen der Zeitsynchronisierung in verteilten Systemen und kennen dazugehörige Algorithmen Die Studierenden kennen die Grundlagen von verteilten DBMS und verteilten Transaktionen. Die Studierenden kennen die Herausforderungen in Zusammenhang mit dem Aufbau von hoch-verfügbaren und skalierbaren Systemen und sind in der Lage geeignete Muster anzuwenden um Verfügbarkeit und Skalierbarkeit von Systemen zu erhöhen Die Studierenden kennen die Grundlagen der Konsistenzmodelle und den Trade-off zwischen Konsistenz und Verfügbarkeit / Skalierbarkeit (CAP-Theorem) Die Studierenden kennen die Grundlagen von NoSQL Datenbanken, deren Vor-/Nachteile und sind in der Lage eine passende NoSQL Datenbank für eine bestimmte Applikation auszuwählen Die Studierenden kennen das Map/Reduce Programmiermodell zur Verarbeitung von grossen Datenmengen Die Studierenden kennen die Grundlagen von Objekt- und Komponenten-basierten Systemen Die Studierenden kennen die Grundlagen der reaktiven Programmierung 		
Lerninhalte	Architekturparadigmen für verteilte Systeme, Grid & Cloud Computing, Algorithmen zur Zeitsynchronisierung in verteilten Systemen, verteilte DBMS und verteilte Transaktionen, Konsistenzmodelle, Architekturen von NoSQL-Datenbanken, Objekt-basierte (RMI, Corba) und Komponenten-basierte (JEE) Systeme, Reaktive Programmierung		
Lehr- und Lernmethoden (Fernstudium nach dem Blended-Learning-Konzept)	Selbststudium <ul style="list-style-type: none"> Erarbeiten des Stoffes Lektüre Lernleistungen Semesterarbeit 	Online-Studium <ul style="list-style-type: none"> Forumsdiskussionen Einreichen von Aufgaben Repetitionsaufgaben Online-Feedback, etc. 	Präsenzstudium <ul style="list-style-type: none"> Lehrgespräch Kurzreferate Gruppendiskussionen Präsentationen, etc.
Unterrichtssprache	Deutsch		
Leistungsbewertung	<ul style="list-style-type: none"> Semesterarbeit Schriftliche Semesterprüfung 		
Lehrmittel	[1] Andrew S. Tanenbaum, Maarten van Steen, Verteilte Systeme, München 2008, ISBN 978-3-8273-7293-2		

	<p>[2] Online Inhalt zu Cloud Computing (wird im Moodle hochgeladen)</p> <p>[3] Online Inhalt zur Transaktionsverwaltung (wird im Moodle hochgeladen)</p> <p>[4] Online Inhalt zur Reliability von verteilten Datenbanken (wird im Moodle hochgeladen)</p> <p>[5] Online Inhalt zur Datenreplikation (wird im Moodle hochgeladen)</p> <p>[6] NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Addison-Wesley, 2012. ISBN: 0-321-82662-0 2005. ISBN: 1-447-12415-4.</p> <p>[7] YCSB Benchmarking: https://www.cs.duke.edu/courses/fall13/compsci590.4/838 CloudPapers/ycsb.pdf</p> <p>[8] Reaktives Manifesto: http://www.reactivemanifesto.org/</p> <p>[9] Reaktive Programmierung mit RxJava: http://gotocon.com/dl/goto-aar-2013/slides/BenChristensen_FunctionalReactiveProgrammingWithRxJava.pdf</p> <p>[10] Jürgen Dunkel et al. Systemarchitekturen für verteilte Anwendungen. Client-Server, Multi-Tier, SOA, Event Driven Architecture, P2P, Grid, Web 2.0. Carl Hanser Verlag. Erste Auflage. 978-3446413214</p> <p>[11] Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems 1st Edition. O'Reilly Media. 978-1449373320.</p>
Vorkenntnisse: Modul(e)	OOP, FTOOP, WebG
Anschlussmodul(e)	-
Bemerkungen	-

*Studienniveau	<p>B Basic level course: Modul zur Einführung in das Basiswissen eines Gebiets.</p> <p>I Intermediate level course: Modul zur Vertiefung der Basiskenntnisse.</p> <p>A Advanced level course: Modul zur Förderung und Verstärkung der Fachkompetenz.</p> <p>S Specialised level course: Modul zum Aufbau von Kenntnissen und Erfahrungen in einem Spezialgebiet.</p>
**Typus	<p>C Core course: Modul des Kerngebiets eines Studienprogramms.</p> <p>R Related course: Unterstützungsmodul zum Kerngebiet (z.B. Vermittlung von Vor- oder Zusatzkenntnissen).</p> <p>M Minor course: Wahl- oder Ergänzungsmodul.</p>

1 Stoffplan

Das Modul VSA hat zum Ziel, die Studierenden mit den Grundlagen von verteilten Systemen vertraut zu machen. Dabei werden grundlegende Konzepte, wie z.B. verteilte Algorithmen, verteilte DBMS und Transaktionen, besprochen, sowie brandaktuelle Themen, wie Cloud Computing, NoSQL und Map/Reduce als konkrete Anwendungen der gelernten Konzepte.

Teil des Moduls ist auch eine Semesterarbeit, welche die Studierenden über das gesamte Semester begleitet. Diese bietet eine gute Möglichkeit, die gelernten Konzepte anzuwenden und zu vertiefen. Diese wird 40% zur Gesamtnote beitragen.

Der Stoff wird entsprechend in folgende Teile aufgeteilt:

Grundlagen, Ziele und Architekturen verteilter Systeme

- Definition verteilter Systeme
- Architektur Paradigmen
- Grid & Cloud Computing

Verteilte Algorithmen

- Uhrensynchronisation
- Physische Uhren
- Logische Uhren
- Leader Election
- Gegenseitiger Ausschluss

Verteilte DBMS und verteilte Transaktionen

- Skalierbarkeit
- Fehlertoleranz
- Verteilte DBMS
- Verteilte Transaktionen
- 2PC, 3PC
- Replikation

NoSQL-Datenbanken & Map/Reduce

- Konsistenzmodelle
- CAP Theorem
- Ziele und Aufbau von NoSQL-Datenbanken
- Large-Scale Datenverarbeitung mit Map/Reduce

Middleware & Komponenten-basierte Architekturen

- Verteilte Objekt-basierte Systeme (CORBA, RMI)
- Komponenten-basierte Systeme (JEE)
- Web-basierte Systeme
- Message-Oriented Middleware

Reaktive Programmierung

- Konzepte
- RxJava