

Verteilte Systeme

Jürgen Kleinöder

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)
www4.cs.fau.de

Sommersemester 2013

http://www4.cs.fau.de/Lehre/SS13/V_VS



Überblick

1 Organisatorisches

- 1.1 Kontakt
- 1.2 Inhalt der Veranstaltung
- 1.3 Vorlesungsbetrieb
- 1.4 Prüfungsmöglichkeiten



Kontakt

Dozenten

- Jürgen Kleinöder
- Tobias Distler

Übungsbetreuung

- Tobias Distler
- Christopher Eibel
- Timo Hönig
- Klaus Stengel



Zielsetzung

- Beantwortung der Frage: „Was macht ein verteiltes System eigentlich zu einem Verteilten System?“
 - Ein verteiltes System ist mehr als die Vernetzung von Rechnern
 - Rechnernetze sind nicht Thema der Veranstaltung
- Vermittlung der Grundlagen und der elementaren Problemstellungen Verteilter Systeme
- Verteilte Systeme aus „Systemsicht“ (Abstraktionen, Ressourcen, ...)
 - Erweiterung des Betriebssystembegriffs in Richtung Verteilte Systeme



Lernziele

- Verständnis der grundlegenden Problemstellungen und ihrer Lösungsansätze
- Einordnung der verschiedenen Kommunikationskonzepte und -mechanismen
- Kennenlernen gängiger Systemarchitekturen
- Tieferes Verständnis für ausgewählte Problemstellungen in verteilten Systemen, z.B.:
 - Behandlung von Zeit
 - Synchronisation in verteilten Systemen
 - Unterschiedliche Sichtweise von Prozessen auf den „aktuellen“ Systemzustand
 - Umgang mit Fehlern
 - Replikation



... Lernziele

- Praktische Erfahrungen mit der Lösung ausgewählter Problemstellungen
 - Entwicklung eines Fernaufrufsystems von Grund auf
 - Realisierung ausgewählter Algorithmen für
 - verteilte und
 - fehlertolerante Systeme



Inhalt – Vorlesung (Teil A: Grundlagen von VS)

- Bestandsaufnahme, Beispiele Verteilter Systeme, Problembereiche
- Eigenschaften
 - Physikalische/logische Verteiltheit
 - Heterogenität, Nebenläufigkeit, Fehlerverarbeitung
 - Sicherheit, Offenheit, Skalierbarkeit, Transparenz
- Architekturen Verteilter Systeme
- Interprozesskommunikation und Fernaufrufe
 - Nachrichtenaustausch
 - IPC-Semantiken und -varianten
 - Fernaufrufe – Kommunikation und Semantikaspekte
 - Fernaufrufe – Parameterübergabe, Nachrichtenerstellung, Realisierungsaspekte



... (Teil B: Middleware und Verteilte Algorithmen)

- Verteilte Anwendungen und Middleware
- Zeit in Verteilten Systemen
 - Logische Uhren
 - Uhrensynchronisation
- Verteilte Algorithmen
 - Synchronisation und gegenseitiger Ausschluss
 - Wahlverfahren
 - Multicast Kommunikation
- FT-CORBA
 - Middleware und Replikationskonzepte
- Verteilte Algorithmen für fehlertolerante Programme
 - Unzuverlässige und zuverlässige Verbindungen
 - Ausfallerkennung
 - Synchrone/asynchrone Systeme



Inhalt – Übung

- Teil A: Fernaufrufsystem
 - Implementierung eines Java-RMI-ähnlichen Systems
 - RMI als Anwender ausprobieren
 - Serialisierung in Java
 - Threads und Synchronisierung in Java
 - (Dynamische) Generierung von Proxies
 - Rückruf/Callback
 - RPC-Semantiken
- Teil B: Verteilte Algorithmen
 - Basisabstraktionen für verteilte Algorithmen
 - Implementierung einfacher verteilter Algorithmen



Vorlesungsbetrieb

- Vorlesungstermin
 - Montag 12 - 14 oder Mittwoch 12 - 14 (noch festzulegen)
 - Ort: 0.031-113
- Foliensatz
 - Ausdrucke werden in der Vorlesung zur Verfügung gestellt
 - außerdem über die WWW-Seite der Veranstaltung abrufbar



Vorlesungsbetrieb

- Rückmeldungen und Fragen
 - Geben Sie uns Rückmeldungen über den Stoff.
Nur so kann eine gute Vorlesung entstehen und gut bleiben.
 - Stellen Sie Fragen!
 - Machen Sie uns auf Fehler aufmerksam!
 - Nutzen Sie auch außerhalb der Vorlesung die Möglichkeit, uns anzusprechen:
persönlich (Zimmer 0.041 / 0.043 im RRZE-Gebäude, Martensstr. 1)
E-Mail {distler, jk}@cs.fau.de



Übung

- Übungstermin
 - Übungsbeginn ist in der Woche ab **22.04.2013**
 - Tafelübung: Dienstag, 12:30 - 14:00 Uhr, Raum 0.031-113
 - Rechnerübung: Dienstag, 14:00 - 16:00 Uhr, Raum 02.155-113
- Inhalt der Tafelübungen
 - Ergänzende und vertiefende Informationen zur Vorlesung
 - Hilfestellungen zu den Übungsaufgaben
 - Klärung von Fragen
 - Anmeldung zu den Übungen: Web-Anmeldesystem Waffel
<https://waffel.informatik.uni-erlangen.de>



Prüfungsmöglichkeiten

- Bachelor und Master Informatik
 - 5 ECTS- oder 7,5 ECTS-Modul in der Vertiefung Verteilte Systeme und Betriebssysteme
- Bachelor IuK
 - 5 ECTS-Modul als „Wahlpflichtmodul aus Katalog für IuK“
- Master IuK
 - 5 ECTS- oder 7,5 ECTS-Modul als „Wahlpflichtmodul aus INF“ in den Schwerpunkten
 - Eingebettete Systeme
 - Kommunikationsnetze
 - Realisierung von Informations- und Kommunikationssystemen
 - Übertragung und Mobilkommunikation
- Bachelor und Master Mechatronik
 - 5-ECTS-Modul in der Modulgruppe „(Verteilte) Eingebettete Systeme“
- Wahlmodul in verschiedenen anderen Studienfächern



Prüfungsmöglichkeiten – Modulvarianten

- 5 ECTS: Vorlesung + Übung
 - erfolgreiche Bearbeitung der abzugebenden Übungsaufgaben
 - mündliche Prüfung über Vorlesungs- und Übungsstoff
- 7,5 ECTS: Vorlesung + erweiterte Übung
 - erfolgreiche Bearbeitung der abzugebenden Übungsaufgaben
 - erfolgreiche Bearbeitung der Zusatzaufgaben
 - mündliche Prüfung über Vorlesungs- und Übungsstoff



Literatur



George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair.
Distributed Systems: Concepts and Design.
Addison Wesley, fifth edition, 2011.

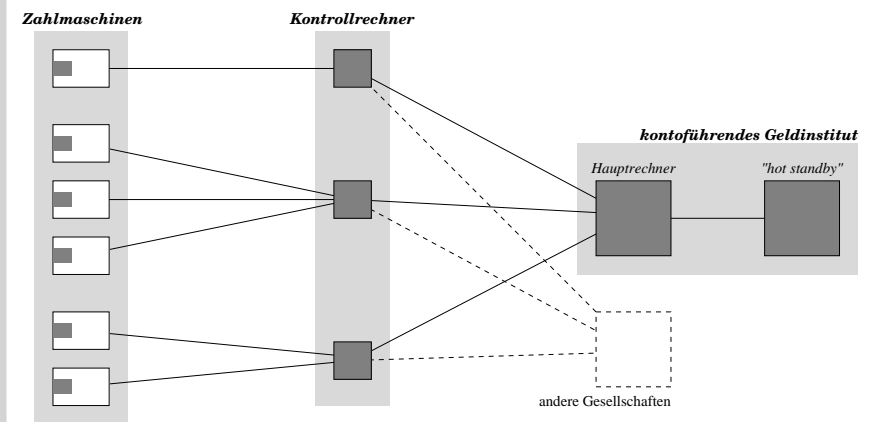
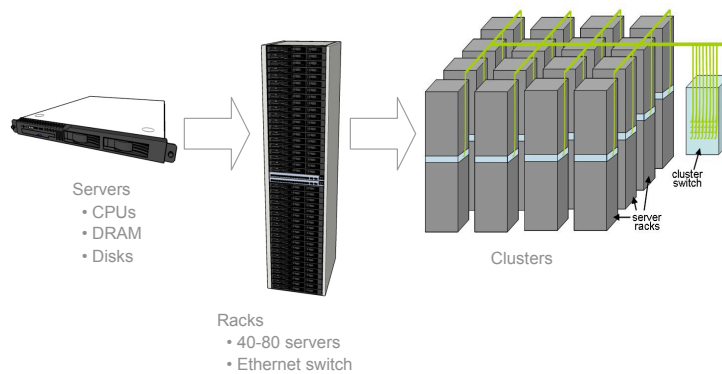


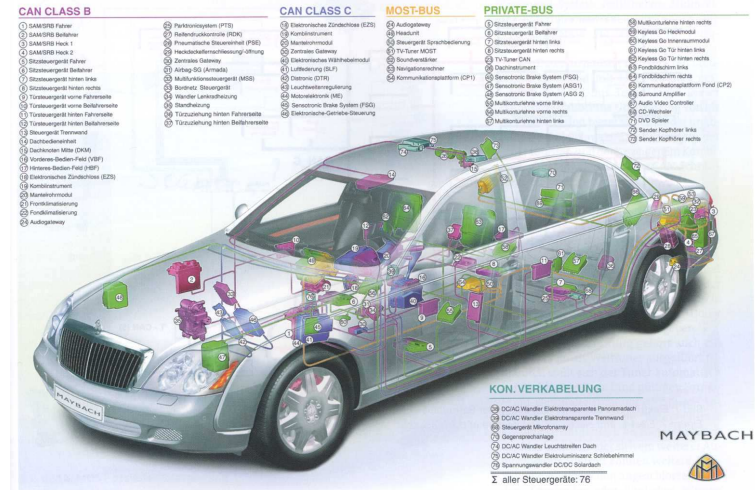
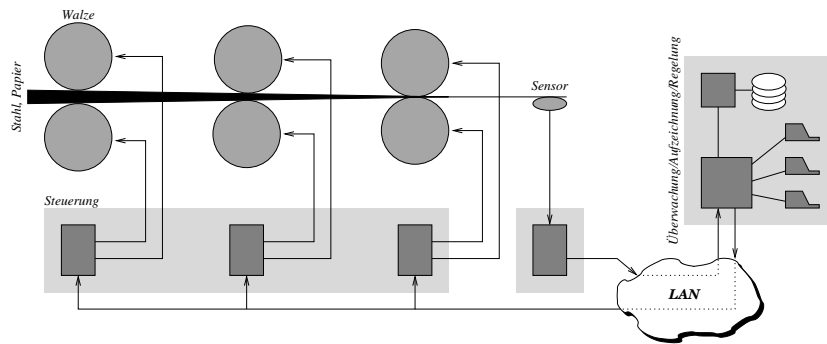
Andrew S. Tanenbaum and Maarten Van Steen.
Distributed Systems: Principles and Paradigms.
Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.



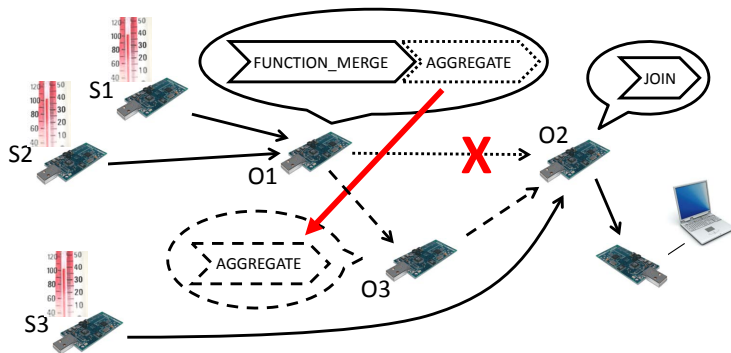
2 Bestandsaufnahme

- 2.1 Beispiele von verteilten Systemen
- 2.2 Anwendungsszenarien
- 2.3 Vorteile
- 2.4 Problembereiche





Quelle: [2]



- Ein Benutzer an einem Ort
 - hoch-parallele Anwendung
 - effizientes Rechnen auf vielen Rechnern
 - Beispiele: Simulationen, meteorologische oder aerodynamische Berechnungen
- Ein Benutzer an mehreren Orten
 - Verwendung verschiedener Rechner
 - Wunsch nach homogener Arbeitsumgebung
 - Beispiele: zentrale Datenhaltung (Fileserver), zentraler Terminkalender



Anwendungsszenarien (2)

- Viele Benutzer an vielen Orten
 - Effizienz, Lastverteilung, Verfügbarkeit
 - Überwindung der Ortsgrenzen
 - Beispiele:
 - Virtuelle Welten, Mehrbenutzer-Spiele
 - Chat, E-Mail, Videokonferenz
 - E-Commerce, CSCW, weltweite Produktentwicklung



Verteilte Systeme: Merkmale

- Mehrere, unabhängige Rechner
 - können unabhängig voneinander ausfallen
- Verbunden durch ein Netzwerk
 - Interaktion nur durch Nachrichtenaustausch möglich
 - Netzwerk unzuverlässig, mit variablen Nachrichtenverzögerungen, moderate Übertragungsgeschwindigkeit im Vergleich zu Multiprozessor-/Multicoresystemen

⇒ Unterschied zu Parallelrechnern
- Kooperation der Knoten
 - Beteiligte Knoten interagieren, um gemeinsam eine Aufgabe zu lösen oder einen Dienst anzubieten

⇒ Unterschied zu einem Rechnernetz



Vorteile

- Rechenleistung vor Ort
 - persönlicher Rechner statt Anschluss an Zentralrechner
 - inhärent verteilte Anwendungen
- Effizienz / Rechenleistung / Skalierbarkeit
 - Einfacher Einsatz mehrerer Rechner
 - gutes Verhältnis Kosten zu Effizienz
- Verfügbarkeit und Zuverlässigkeit
 - Redundante Auslegung von Komponenten
 - Gesamtsystem bleibt auch bei Ausfall einzelner Komponenten verfügbar



Problembereiche (1)

lokal ⇒ entfernt

- Im Falle entfernt ausgelegter Interaktionen sind mehr Fehlerarten möglich als im Falle nur lokal ausgelegter Interaktionen.

direkte ⇒ indirekte Bindung

- Konfigurierung wird zu einem dynamischen Vorgang und erfordert Bindungsunterstützung zur Laufzeit.

sequentielle ⇒ nebenläufige Ausführung

- Nebenläufigkeit durch Parallelität erfordert Mechanismen zur Koordinierung der Aktivitäten.



Problembereiche (2)

synchrone \Rightarrow asynchrone Interaktion

- Verzögerungen durch die Kommunikation erfordern Unterstützung für asynchrone Interaktionen und zur Fließbandverarbeitung (*pipelining*).

homogene \Rightarrow heterogene Umgebung

- Interaktionen zwischen entfernten Systemen erfordern eine gemeinsame Datenrepräsentation.

einzelne Instanz \Rightarrow replizierte Gruppe

- Replikation kann Verfügbarkeit (*availability*) und/oder Zuverlässigkeit (*dependability*) bereitstellen, erfordert aber auch Maßnahmen zur Konsistenzwahrung.



Problembereiche (3)

fester Platz \Rightarrow Wanderung

- Die Lage entfernter Schnittstellen (zu Funktionen, Objekten, Komponenten) kann sich zur Laufzeit ändern.

einheitlicher \Rightarrow zusammengeschlossener Namensraum

- Die Namensauflösung muss (ggf. bestehende) Verwaltungsgrenzen zwischen verschiedenen entfernten Systemen reflektieren.

gemeinsamer \Rightarrow zusammenhangloser Speicher

- Mechanismen des gemeinsamen Speichers sind nicht (oder nur sehr eingeschränkt) im großen Maßstab anwendbar.



Verteilte Systeme: Anmerkungen und Definition

Leslie Lamport

A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.

Paulo Verissimo

If you do not need a distributed system, do not distribute.

\Rightarrow Fehlertoleranz von verteilten Systemen ist eine sehr wichtige Eigenschaft, die auch heute noch in vielen Systemen fehlt!

Definition von Andrew Tanenbaum

Ein verteiltes System ist eine Kollektion unabhängiger Computer, die den Benutzern als ein Einzelcomputer erscheinen.



Referenzen



Architecture Projects Management Ltd.
ANSA: An Engineer's Introduction to the Architecture.
Technical Report TR.03.02, Castle Hill, Cambridge, UK, November 1989.
<http://www.ansa.co.uk/ANSATech/89/TR0302.pdf>.



DaimlerChrysler AG.
Der neue Maybach.
ATZ/MTZ Sonderheft, page 125, September 2002.



Tony King.
Pandora: An Experiment in Distributed Multimedia.
In *Proceedings of Eurographics '92*, Cambridge, UK, September 1992.
<http://www.uk.research.att.com/pandora.html>.

