

Verteilte Systeme

Wintersemester 2002
Universität Trier
Peter Sturm

Abstract

In einem verteilten System kommunizieren mehrere Prozesse durch den Austausch von Nachrichten. Ausführungsplattform sind dabei viele, über ein Kommunikationsnetz verbundene Computer. Die Vorteile einer verteilten Realisierung von Systemsoftware und Anwendungsprogrammen sind:

- Inkrementelle Erweiterbarkeit
- Leistungssteigerung durch Parallelarbeit
- Erhöhung der Ausfallsicherheit durch Redundanz

Die Realisierung dieser Systeme wird jedoch durch eine Reihe prinzipieller Beschränkungen wie z.B. dem Fehlen eines globalen Zeitbegriffs und einer globalen Systemsicht erschwert. In dieser Vorlesung werden die Möglichkeiten und Grenzen verteilter Systeme vorgestellt. Vorlesungsinhalte:

- Eigenschaften von Computernetzen
- Kommunikationsprotokolle (IPv4, IPv6, ...)
- Verteilte Algorithmen
- Fehlertoleranz
- Lastverteilung
- RPC und Client/Server-Systeme
- Verteilte Ausführungsplattformen (PVM, MPE, DCOM, CORBA, ...)
- Middleware
- Alternative Ansätze (Verteilter gemeinsamer Speicher, ...)

Organisatorisches

Vorlesungstermine

- Dienstags, 12.30 bis 14.00 Uhr, Raum V302
- Freitags, 10.00 bis 11.30 Uhr, Hörsaal HS10

Übungen

- Donnerstags, 12.30 bis 14.00 Uhr, Raum E 45

Folienkopien

- Unterlagen ab 1998 überwiegend aktuell
- Updates meist unmittelbar vor der Vorlesung erhältlich

<http://www.syssoft.uni-trier.de/systemsoftware>

Fragen, Kritiken, Wünsche

- Mündlich an mich oder
- Email: sturm@uni-trier.de

Noch Fragen, z.B. bzgl. Schein, Prüfung u.ä.?

Verteilte Systeme, Winter 2002/03

Folie 3

Was ist ein verteiltes System?

Rechnernetz

- Mono- und Multiprozessorsysteme
- Kommunikationsnetzwerk

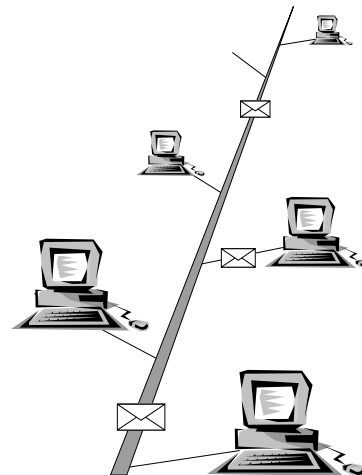
Kein gemeinsamer Speicher !

Senden und Empfangen von Daten

- Nachrichten

Eigenschaften

- Inkrementelle Erweiterbarkeit
- Hohe Nominalleistung



Wieviele Rechner mit jeweils 100 MIPS bilden ein Rechnernetz mit einer scheinbaren Instruktionszeit von 1 Pikosekunde (10^{-12} s)?

Was würde ein Monoprozessor mit einer solchen Taktfrequenz kosten?

Verteilte Systeme, Winter 2002/03

Folie 4

Nachrichtenbasierte Kommunikation

„Größe“ von Sender und Empfänger

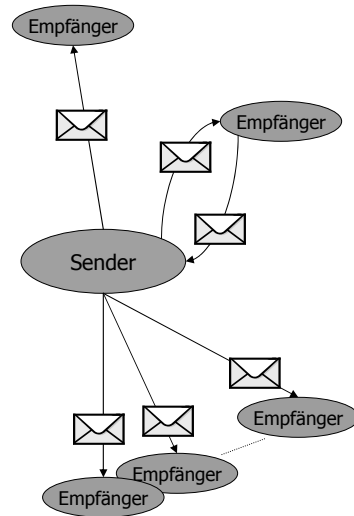
- Prozeß
- Thread
- Komponente (DCOM, .NET, Corba, JavaBean, ...)
- Objekt

Nachricht

- Untypisiert (n Bytes)
- Enthält Typinformation
- Enthält ausschließlich Daten (Data Shipping)
- Enthält auch Code (Data Shipping)

Vielfältige Interaktionsmuster

- Mitteilung, Auftrag
- Synchron, Asynchron
- Direkt oder Indirekt (Port, Mailbox, ...)
- Unicast, Multicast, Broadcast
- ...



Verteilte Systeme, Winter 2002/03

Folie 5

Vorteile und herausfordernde Probleme

Vorteile

- Leistungssteigerung durch Parallelarbeit
- Erhöhte Verfügbarkeit
- Inhärent verteilte Anwendungen

Probleme

- Konsistente globale Zustände
- Fehlen einer globalen Uhr
- Komplexität

Eigenschaften

- Übertragungsgeschwindigkeit ist endlich
- Nachrichten sind bei der Ankunft bereits veraltet



Verteilte Systeme, Winter 2002/03

Folie 6

Vorteil 1: Leistungssteigerung durch Parallelarbeit

Erzielbarer Leistungsgewinn

- Rechenzeit auf Monoprozessor: $T_1(n)$
- Rechenzeit auf k Monoprozessoren: $T_k(n)$
- Maximaler Speedup: $\frac{T_1(n)}{T_k(n)} \leq k$

Wie definiert man die Rechenzeit bei k Monoprozessoren?

Warum ist der Speedup meist ein gutes Stück kleiner als k?

Partitionierung der Aufgabe

- Möglichst hoher Rechenbedarf innerhalb einer Partition
- Möglichst geringer Kommunikationsbedarf zwischen Partitionen

Genügend Rechenlast

Verteilte Systeme, Winter 2002/03

Folie 7

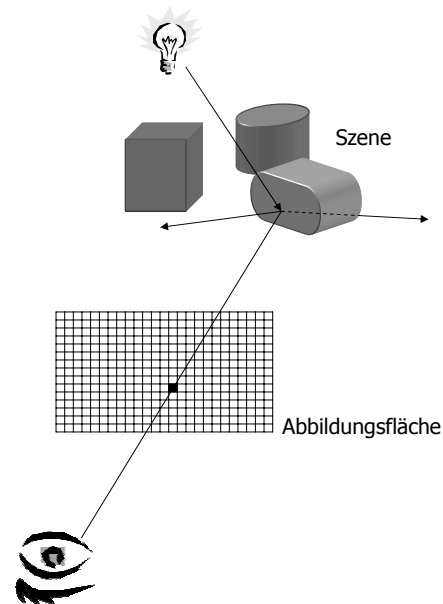
Beispiel: Verteiltes Raytracing

Identische Szenenbeschreibung auf allen Rechnern

- u.U. umfangreiche Daten
- Zentraler Ursprung
- Effiziente Verteilung

Sinnvolle Partitionierungsansätze?

Erreichbarer Speedup?



Verteilte Systeme, Winter 2002/03





Folie 8

Partitionierung entscheidet über Speedup

Vergleichbarer Rechenaufwand in jedem Abschnitt

Speedup k prinzipiell erreichbar



Wie ermittelt man eine geeignete Partitionierung?

Rechner 1	Rechner 2
	
Rechner 3	Rechner 4
	

Starkes Ungleichgewicht beim Rechenaufwand

Unter Umständen Speedup < 1

- Zusätzlicher Initialisierungsaufwand
- Zusammenfassen der Einzelresultate

Verteilte Systeme, Winter 2002/03 Folie 9

Vorteil 2: Erhöhte Verfügbarkeit (Zuverlässigkeit)

Verteiltes System = Redundanz

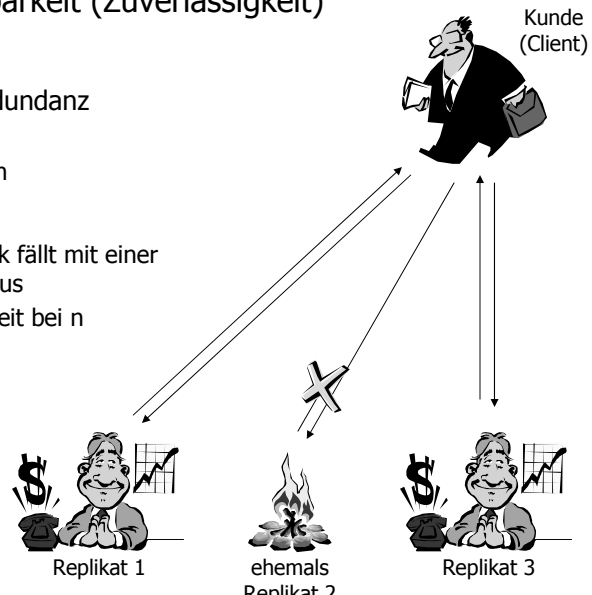
- Rechnern
- Kommunikationsnetzen

Grundidee

- Einzelne Komponente k fällt mit einer Wahrscheinlichkeit p aus
- Ausfallwahrscheinlichkeit bei n Replikaten von k :

$$p^n < p$$

Voraussetzung für $p^n < p$?



Kunde (Client)

Replikate 1 ehemals Replikate 2 Replikate 3

Verteilte Systeme, Winter 2002/03 Folie 10

Beispiel: Zuverlässiger Bank-Server

Zustandsändernde Transaktionen

Realisierung 1:
Wir vertrauen der Software!?

Realisierung 2:
Toleranz gegenüber Software-Fehlern?

Folie 11

Vorteil 3: Inhärent verteilte Anwendungen

Räumlich verteilte Partner

Trivial: Asymmetrische Lösung

- Zentraler Server
- „Dump Clients“

Vor- und Nachteile?

Symmetrische Lösung

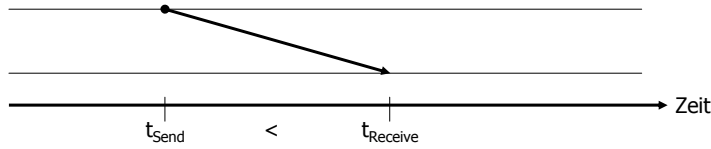
- Zustände a priori verteilt
- Nebenläufige Kontrollflüsse isolierbar

Vor- und Nachteile?

Folie 12

Probleme

Endliche Ausdehnungsgeschwindigkeit der Information



- Zeitsynchronisation schwierig
- Veraltete Information

Eigenschaften des Kommunikationsnetzwerkes

- Schwankungen in der Latenz
- Verluste, Vertauschungen, Duplikate

Implizite und explizite Abhängigkeiten

Verteilte Systeme, Winter 2002/03

Folie 13

Problem: Konsistenz globaler Zustände

Beispiel: Terminierung in einem verteilten Aktorensystem

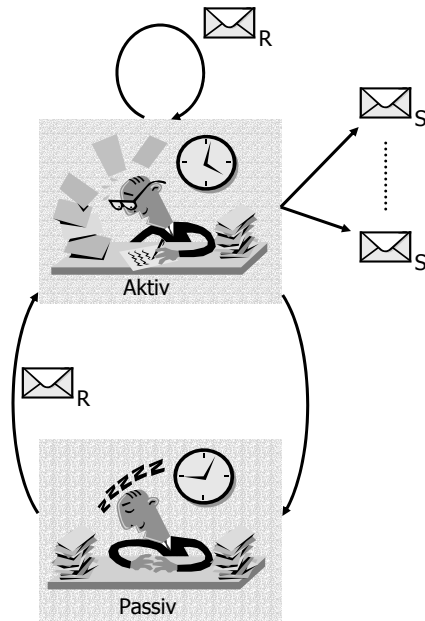
Aktoren kommunizieren nur durch Nachrichten

Zustände eines Aktoren

- Aktiv
- Passiv

Was heißt "Terminierung des verteilten Aktorensystems"?

Lösungsvorschlag?



Verteilte Systeme, Winter 2002/03

Folie 14

Problem: Fehlen einer globalen Zeit



Wie schon Einstein sagte: „Alles ist relativ“

Bedeutung der Zeit

- Vergleichbarkeit und Ordnung von Ereignissen
- Vielfach versteckte Abhängigkeiten

Gleichzeitigkeit

Konsequenzen

- Naiv: Neue Physik muß her
- Pragmatisch: Hinreichend präzise Uhrensynchronisation
- Theoretisch: Schwächere Zeitbegriffe (Kausalität)

Warum würde es bei einer gemeinsamen globalen Zeit z.B. die Problematik konsistenter verteilter Zustände nicht geben?

Single-System-Image: Realität oder Utopie?

Rechnernetz erscheint wie ein virtueller Multiprozessor

- mit allen Vorteilen
- ohne die Nachteile



=



Leistungen der verteilten Systemsoftware?

Aktuelle Tendenzen

Nachrichtenlaufzeit war bisher dominant

- Signifikante Verzögerungen der beteiligten Prozesse
- Leistungssteigerungen schwierig
- Probleme verteilter Systeme unmittelbar spürbar



Ultraschnelle Netze

Der Hauptspeicher des Nachbarrechners ist näher als meine lokale Platte?

- Probleme verteilter Systeme bis zu einer gewissen Ausdehnung vernachlässigbar
- Cluster

Component-Ware

- DCOM, COM+, .NET, Corba, EJB, ...

... viele andere spannende Sachen

Zusammenfassung

Gute Gründe dafür

- Leistungssteigerung
- Zuverlässigkeit
- Natürliches Modell

„Dagegen“-Gründe sind lösbar

- Konzepte
- Pragmatische Ansätze
- Unterstützung
- Vereinfachende Modelle

Große Zukunft



Begleitende Literatur

G. Coulouris, J. Dollimore, T. Kindberg
Distributed Systems - Concepts and Design
3. Auflage, Addison-Wesley, 2000

Greg Gagne, Peter Galvin, Peter B. Galvin, Avi Silberschatz
Applied Operating System Concepts
John Wiley & Sons, 1999

Andrew Tanenbaum, Maarten Van Steen
Distributed Systems
Prentice Hall, 2002

Spezialliteratur im Verlauf der Vorlesung