

Halbleitertechnologie: Leistung und Kosten

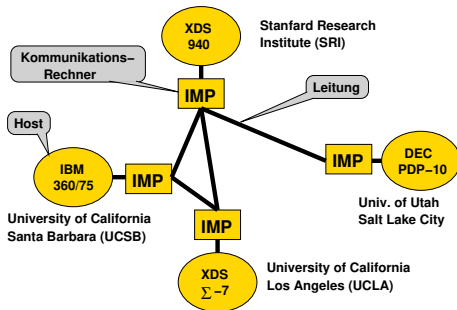


- Speicherchips:
 - ▶ 1973: 4 kBit
 - ▶ 1985: 64 kBit
 - ▶ 1998: 64 MBit
 - ▶ 2008: 16 GBit
 - ▶ 2018: 128 GBit (Samsung DDR4 DRAM)
- Gesetz von Moore (1965): Alle zwei Jahre (18 Monate) verdoppelt sich die Zahl der Transistorfunktionen auf der gleichen Grundfläche.
- Entwicklung der Kosten je Transistorfunktion auf ca. 1/10 alle vier Jahre.
- Immer wieder wurde das Ende des Gesetzes vorausgesagt.
- Neuere Technologien: Z-RAMs, MRAMs, FeRAMs (nicht-flüchtig),
...

Vom ARPANET zum Internet



- 1958** Gründung der Advanced Research Projects Agency (ARPA) durch US Dept of Defense (DoD) als Reaktion auf Sputnik
- 1968** Idee des „Internet“ als „tool to create critical mass of intellectual resources“ (Licklider, Taylor)
Hauptplaner: *Vinton Cerf, Bob Kahn*
- 1969** erstes funktionsfähiges Netz, gemietete 50 kBit/sec Leitungen, Interface Message Processors von BBN (RFC 1)



Vom ARPANET zum Internet (2)



1972 erste öffentliche Demo (remote login)

- Network Control Protocol (NCP) als Protokoll (Menge von Regeln für die Kommunikation)
- Hauptnutzung: Terminal-Sitzungen, Dateitransfer, Electronic Mail

1974 Grundzüge der TCP/IP-Protokolle in Papier von Cerf/Kahn
IP=Internet Protocol, TCP=Transmission Control Protocol,
Standardisierung in den Folgejahren

1982 Übergang zu den heutigen Internet Protokollen TCP/IP Version 4

ab 1983 Verbreitung von TCP/IP durch Berkeley UNIX 4.2 BSD, freie Verfügbarkeit des Quellcodes

1986 Erstes Treffen der Internet Engineering Task Force (IETF) in San Diego

Das World-Wide Web (WWW)



- ab 1970 Arbeiten zu Hypertext-Systemen (durch Zeiger verbundenes verteiltes Geflecht von Knotendokumenten mit einfachen Navigationsmöglichkeiten) von Ted Nelson (Project Xanadu)
- 1990 Vorschlag für ein Hypertext-Projekt am CERN in Genf durch Tim Berners-Lee und Robert Cailliau: Wiege des World-Wide Web
- 1991 Entwicklung einer ersten Version auf NeXT-Rechner
Präsentation auf Hypertext-Konferenz
- 1992 Herausgabe einer freien Version von Web-Server und Browser (Unix-basiert) durch CERN, Ende des Jahres: weltweit ca. 50 Web-Server
- 1993 Marc Andreessen, Eric Bira (NCSA, Univ. of Illinois) geben erste Version des Mosaic Browsers heraus, gründen später Netscape
- 1994 Für Microsoft ist WWW noch kein Thema. Bill Gates: „... *an Internet Browser is a trivial piece of software. There are at least 30 companies that have written creditable Internet browsers, so that's nothing...*“
- Ende '95 Microsoft greift ein...

Ubiquitous Networks



- 1982 An der Carnegie Mellon Universität wird ein Getränkeautomat mit dem Internet verbunden
- 1995 Veröffentlichung der ersten IPv6-Spezifikation
- 1996 Hewlett-Packard und Nokia veröffentlichen mit dem OmniGo 700LX und dem 9000 Communicator erste Smartphone-Vorläufer
- 1997 Kristofer S. J. Pister, Joe Kahn und Bernhard Boser (Berkeley) präsentieren einen Forschungsantrag zum Thema *Smart Dust*
- 1998 Gründung von Google
- 1999 Kevin Ashton prägt den Begriff des *Internet of Things*
- 2001 Wikipedia geht online
- 2004 Gründung von Facebook
- 2007 Apple veröffentlicht das erste iPhone
- 2014 Die IETF-Arbeitsgruppe *CORE* veröffentlicht die erste Spezifikation zum Constrained Application Protocol (CoAP)

Heutige Klassen von Rechensystemen



- Personalcomputer (PC, Desktop), Workstations
- Server, Großrechner (Mainframes)
 - ▶ hochverlässliche Verarbeitung von Massendaten
 - ▶ Hoch- bis Höchstleistungs-Ein-/Ausgabe-Einheiten
 - ▶ Server erbringen Dienstleistungsfunktionen in Rechnernetzen
 - ▶ Mainframes sind z.T. wegen nicht mehr wartbarer Altprogramme erforderlich (Legacy-Systeme)
- Supercomputer
 - ▶ Vielzahl von Prozessoren/Knoten
 - ▶ hohe Verarbeitungsleistung
 - ▶ Beispiel: numerische Berechnungen zur Wettervorhersage
- Embedded Computer (eingebetteter Rechner)
 - ▶ Teil von Maschinen, Geräten oder Anlagen
 - ▶ Rechensystem steht gegenüber der Funktionalität des umgebenden Systems im Hintergrund
 - ▶ Cyber-Physical Systems/Industrie 4.0

Aktuelle Entwicklung



- Heutige Rechner werden zwar immer leistungsfähiger und besitzen ein immer besseres Preis-/Leistungsverhältnis, erreicht wird dies aber nur durch graduelle Verbesserungen bekannter Techniken.
- Ebenen
 - ▶ Prozessoren
 - ★ kürzere Entwicklungszyklen durch verbesserte Design-Werkzeuge
 - ★ Konzentration auf Prozessoren mit Intel-Befehlssatz im Office-Bereich
 - ★ ursprünglich viele μ Controller-Typen in Embedded Systemen, mittlerweile Konzentration auf ARM-Architektur
 - ★ Multicore-Prozessoren
 - ▶ Systeme
 - ★ verstärkter Einsatz von Systemen mit vielen Knoten
 - ★ z.B. Blade-Server, HPC-Cluster
 - ▶ Netzwerke
 - ★ steigende Kommunikationsbandbreiten
 - ★ verschiedenartige Dienstgüteanforderungen
 - ★ mobile Knoten

Aktuelle Entwicklung (2)



- Virtualisierung
 - ▶ Virtuelle Maschinen (VMs)
 - ▶ Speichervirtualisierung (Software Defined Storage)
 - ▶ Virtuelle Netze (Software Defined Networks, SDNs)
- Virtuelle Infrastrukturen (Cloud Computing)
- Internet der Dinge und der Dienste, Industrie 4.0/Industrial Internet
- Big Data

Definitionen



- „A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable.“
Leslie Lamport
- „A distributed system is a collection of autonomous computing elements that appears to its users as a single coherent system.“
Maarten van Steen, Andrew S. Tanenbaum

Grundbegriffe verteilter Systeme



- Enge Kopplung: Zwei Software-Komponenten sollen **eng gekoppelt** heißen, wenn sie durch gemeinsame Nutzung von Betriebsmitteln (Sharing) kommunizieren. Diese können z.B. sein:
 - ▶ gemeinsam genutzte typisierte Objekte
 - ▶ gemeinsam genutzte Speichersegmente
- Lose Kopplung: Zwei Software-Komponenten sollen **lose gekoppelt** heißen, wenn sie durch Nachrichtenaustausch (Message Passing) kommunizieren (höhere Autonomie der Komponenten).
- Analog existieren auf der Ebene der Anwendungsprogrammierung entsprechende Paradigmen, die auf Sharing oder auf Message Passing angelegt sind.

Verteiltes Programm/Verteiltes System

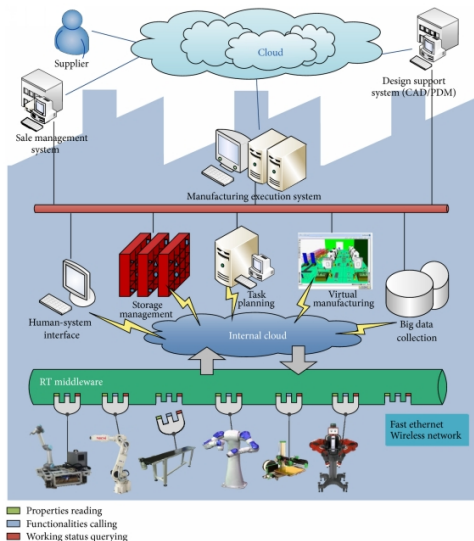


- Ein **verteiltes Programm** besteht aus einer Menge von lose gekoppelten Software-Komponenten, die (durch Nachrichtenaustausch) bzgl. einer gemeinsamen Problemlösung kooperieren.
- Ein verteiltes Programm beinhaltet
 - ▶ einen **verteilten Zustand** (in den jeweiligen Software-Komponenten).
 - ▶ **verteilte Kontrolle/Koordination**, um die gemeinsame Problemlösung zu bewerkstelligen.
- Ein **verteiltes System** ist ein Rechensystem, das ein verteiltes Programm ausführt.

Beispiele verteilter Systeme



- Das Internet
- Peer-to-peer Netze
- Automatisierte Fertigungsanlagen



Transparenzarten



- Transparenz bedeutet, anders als im üblichen Sprachgebrauch, die *Unsichtbarkeit* einer Eigenschaft eines Mehrrechner-Systems.
- Transparenzarten gemäß International Standards Organization (ISO) und Advanced Network Systems Architecture (ANSA)
 - ▶ **Ortstransparenz:** keine Kenntnis des Ortes einer Komponente notwendig, insbesondere ist der Ort nicht Teil des Namens.
 - ▶ **Zugriffstransparenz:** Form des Zugriffs ist unabhängig, ob Komponente lokal oder entfernt ist.
 - ▶ **Migrationstransparenz:** Komponente kann verlagert werden, ohne dass sich die Schnittstelle zu den Nutzern ändert.
 - ▶ **Replikationstransparenz:** Benutzungsschnittstelle ist unabhängig von der Anzahl der Kopien einer Komponente, Verfügbarkeit als qualitative Eigenschaft kann verbessert werden.

Transparenzarten (2)



- Weitere Transparenzarten:

- ▶ **Nebenläufigkeitstransparenz:** Nebenl. Nutzung von Komponenten so, als ob diese privat wären, z.B. durch autom. Sperren, vgl. Datenbanken.
- ▶ **Fehlertransparenz:** Ein eingetretener Fehlzustand wird an der Benutzungsstelle nicht sichtbar, sondern durch Redundanz im unterlagerten System maskiert.
- ▶ **Skalierungstransparenz:** System und Applikationen können sich vergrößern, ohne dass die Systemstruktur oder die Applikationsalgorithmen geändert werden müssen.
- ▶ **Leistungstransparenz:** Neukonfiguration des Systems bei Veränderung der Last, um die Leistung zu verbessern.

Welche Transparenzart ist relevant?



- Transparenz trägt zur Vereinfachung des Umgangs und der Programmierung des Systems bei, da der betreffende Aspekt vom Nutzer des Systems nicht beachtet werden muss.
- Ein verteiltes System sollte möglichst alle angegebenen Transparenzarten verwirklichen, um eine möglichst einheitliche Systemsicht zu erreichen.
- Vollkommene verteilte Systeme, die von allen Aspekten abstrahieren, existieren derzeit nicht.
- Die Unterstützung einzelner Transparenzarten (z.B. Ortstransparenz) ist fortgeschritten.

LOS - NOS - DOS



- Stufen der Entwicklung können an folgenden BS- Architekturklassen festgemacht werden:
- **LOS (Local Operating System):**
übliches Knoten-Betriebssystem ohne Unterstützung für Verteiltheit.
Beispiele:
 - ▶ IBM MVS,
 - ▶ UNIX System III,
 - ▶ DOS, Windows 3.1,
 - ▶ ...

LOS - NOS - DOS (2)



- **NOS (Network Operating System):**

Betriebssystemerweiterung von ev. verschiedenen LOS' für ein Multicomputer-System, um gewisse Funktionen bzgl.

- ▶ Dateisystem,
- ▶ Schutz (Benutzerverwaltung),
- ▶ entfernter Programmausführung

systemweit, mehr oder weniger transparent, bereitzustellen.

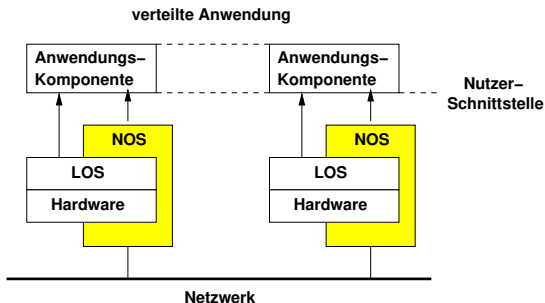
Beispiele:

- ▶ Novell NetWare,
- ▶ Windows for Workgroups,
- ▶ UNIX Yellow Pages (NIS) und Network File System (NFS)

LOS - NOS - DOS (3)



- Prinzipielle Struktur eines Netzwerkbetriebssystems (NOS):



- Die unterlagerten lokalen Knoten-Betriebssysteme (LOS) können gleich oder verschieden sein.
Beispiele:
 - ▶ Netware Client für DOS, NT, ... ; Netware Server ist spezielles BS.
 - ▶ NFS Client für UNIX, NT, ...

LOS - NOS - DOS (4)



• DOS (Distributed Operating System):

Ein verteiltes Betriebssystem ist ein grundständiges BS, das

- ▶ seinen Nutzern eine einheitliche Systemsicht auf ein Multicomputer-System bietet (s.o.),
- ▶ zur Realisierung der Transparenzeigenschaften auf Algorithmen basiert, die unter verteilter Kontrolle unter Austausch von Nachrichten ablaufen.

