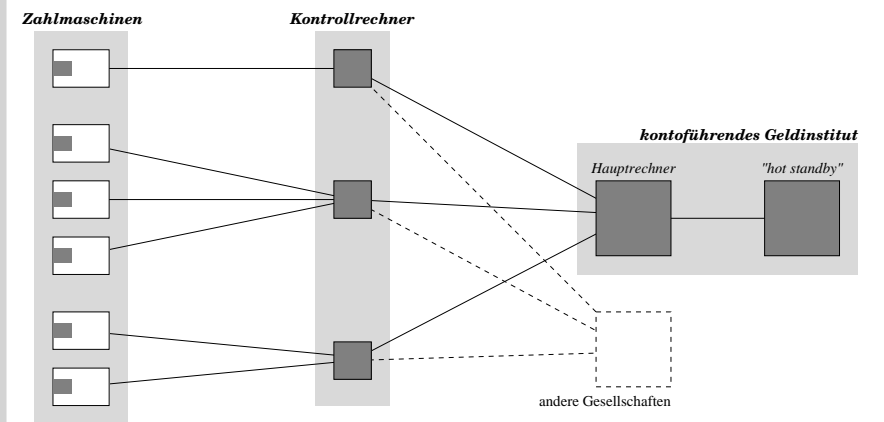
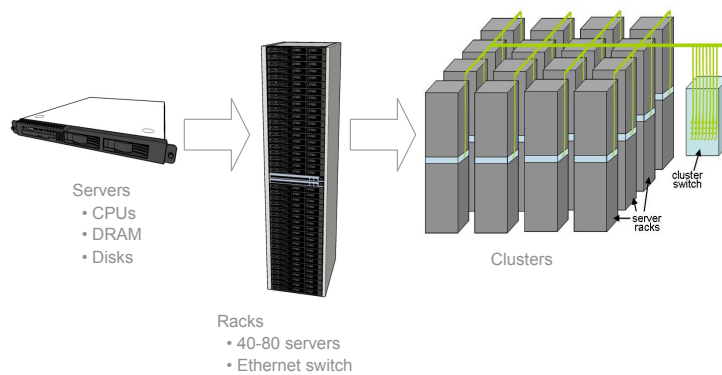
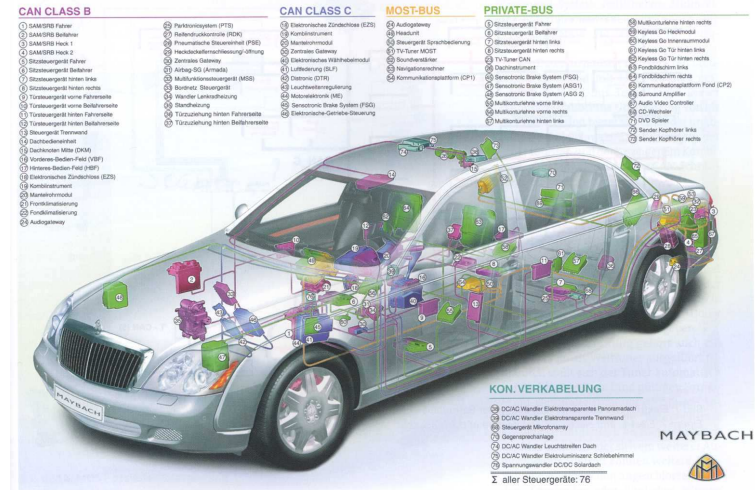
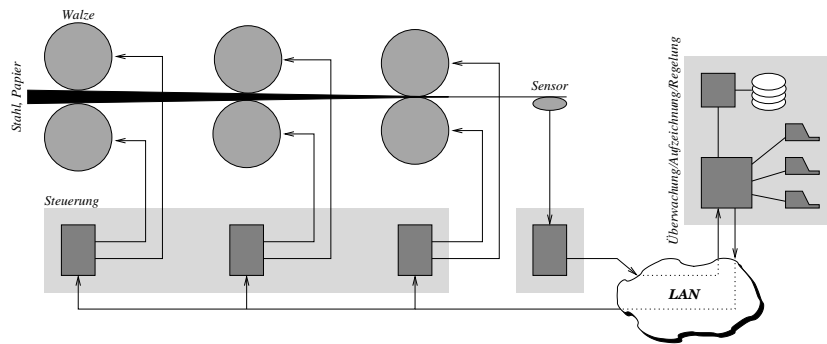


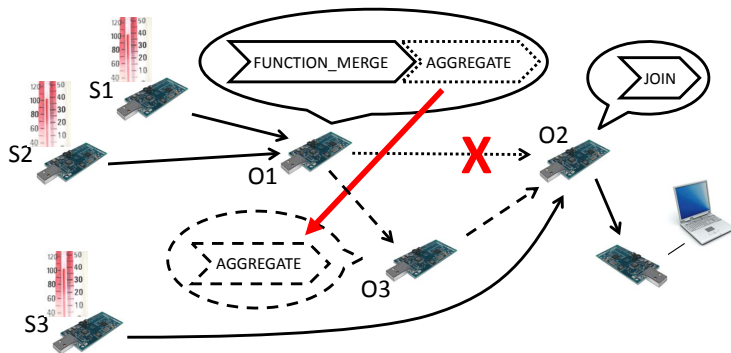
2 Bestandsaufnahme

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





Quelle: [?]



- 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 Orts- und Zeitgrenzen
 - 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 zusammenschlossener 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

- [1] George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. *Distributed Systems: Concepts and Design*. Addison Wesley, fifth edition, 2011.
- [2] Andrew S. Tanenbaum and Maarten van Steen. *Distributed Systems: Principles and Paradigms (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

