

# B Motivation

---

## ■ Forderung an Rechensystem (Hardware und Software):

- ◆ Korrektheit
- ◆ Benutzerfreundlichkeit
- ◆ Performance

## ■ Für moderne Rechensysteme gilt:

- ◆ Performance offensichtlich (z.B. Webperformance)  
Jeder Benutzer hat seine eigene Workstation, PC oder Terminal
- ◆ Oft Mindestperformance erforderlich, z.B. bei Realzeitanforderungen
- ◆ Performance entscheidet oft über die Akzeptanz
- ◆ Performance oft kaufentscheidend

## ■ Maßnahmen zur Verbesserung der Performance

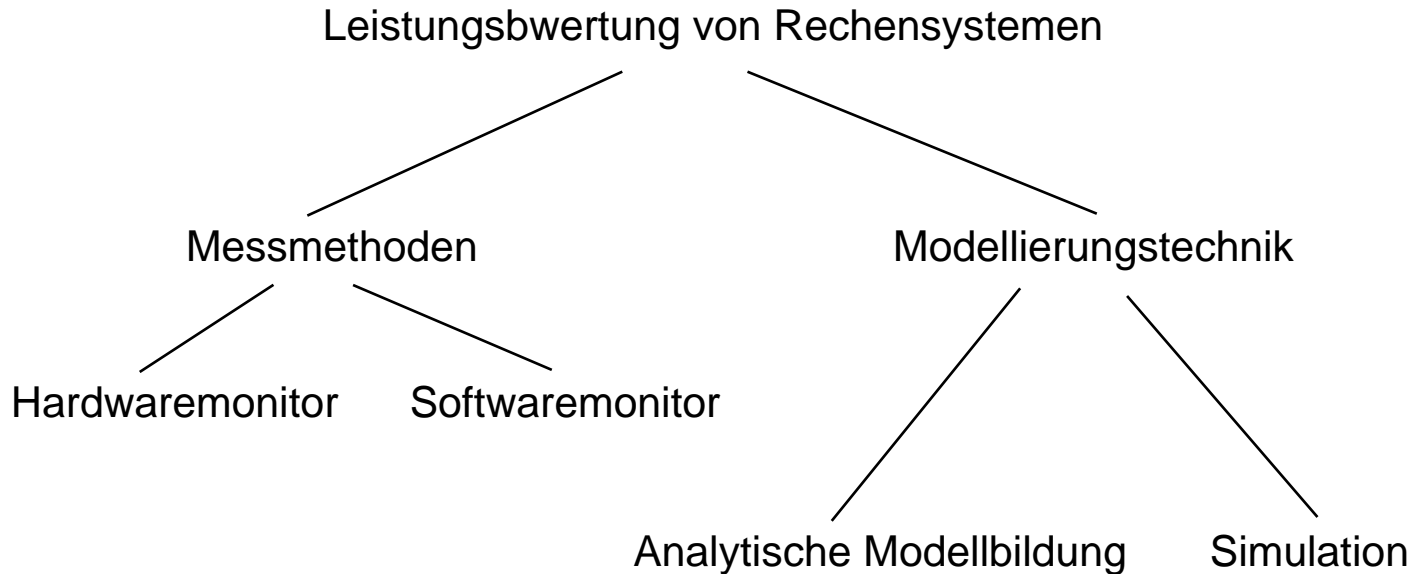
- ◆ Messung, Modellierung und Leistungsbewertung
- ◆ **wichtig bei:** Planung, Entwicklung, Tuning, Vergleich von Rechensystemen

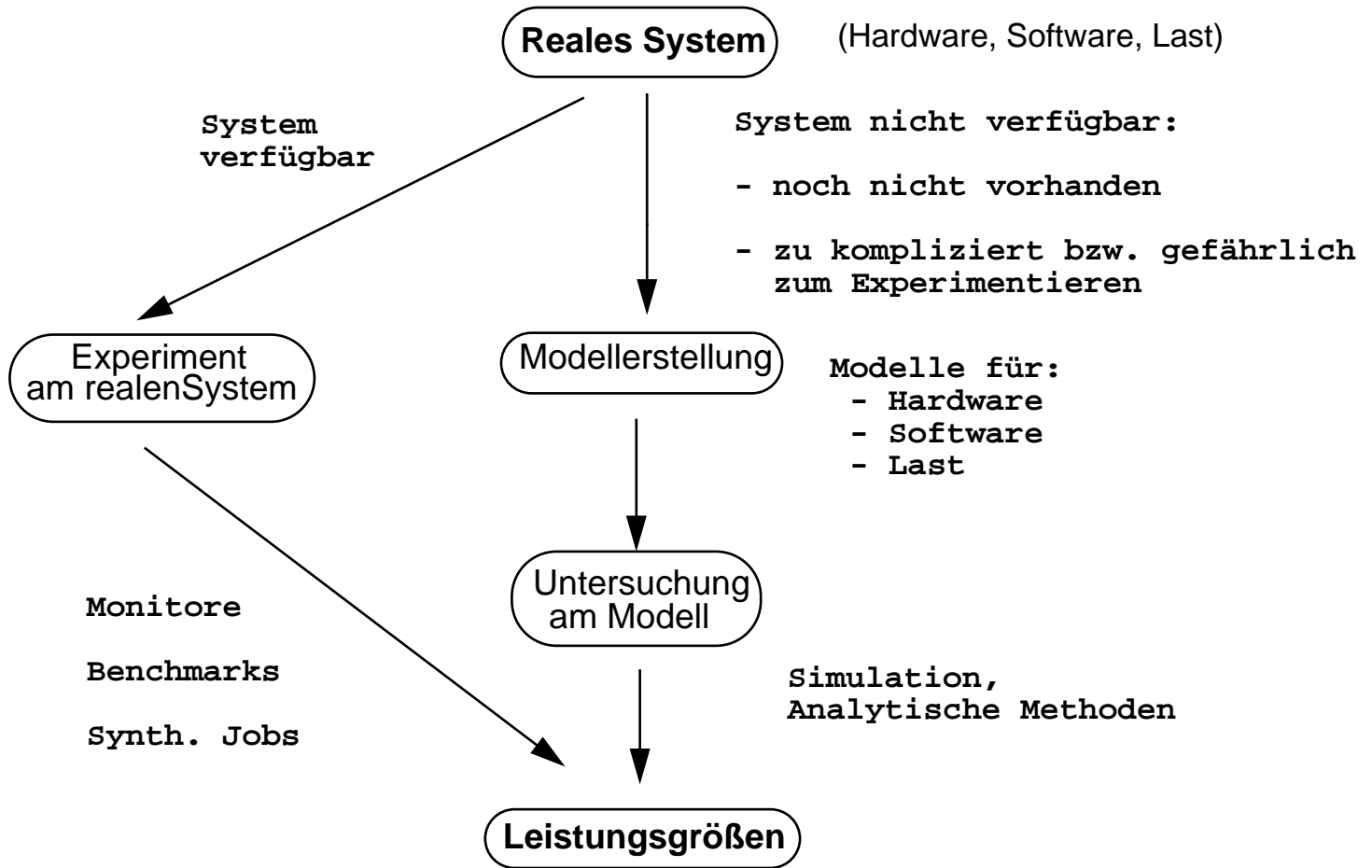
# B.1 Leistungsgrößen

---

- Antwortzeit, Verweilzeit
- Durchsatz
- CPU-Auslastung
- Auslastung von Kanälen und Peripheren Geräten
- Warteschlangenlängen
- Einhaltung von Zeitschranken
- Speed up
- Verfügbarkeit, Zuverlässigkeit
  - Leistung soll bestimmt werden können bei gegebenen Systemparametern
  - Optimale Leistungsgrößen bei variablen Systemparametern

# B.2 Leistungsbewertung im Überblick

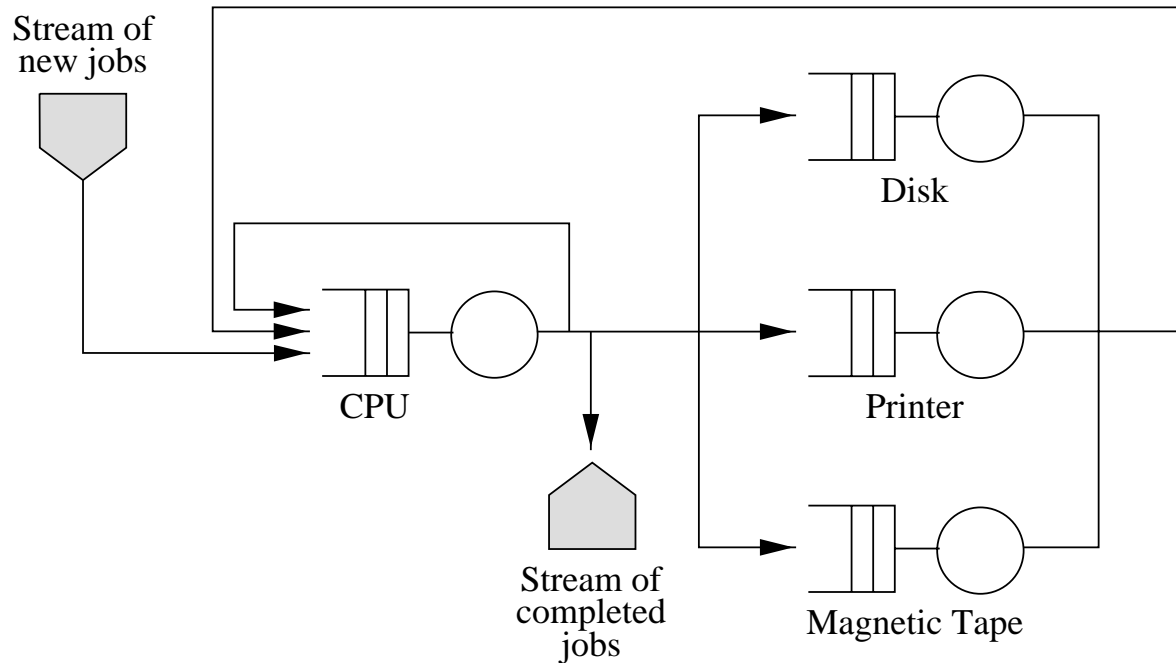




## B.3 Modelltypen

### ■ Warteschlangenmodell:

- ◆ System wird beschrieben durch: WS'n, Bedieneinheiten und Kanten

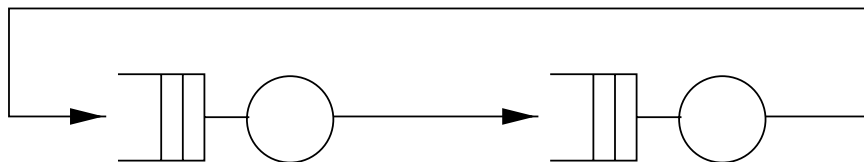
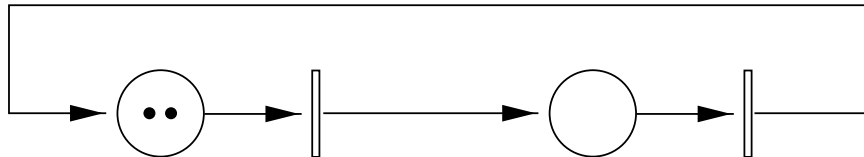


## ■ Petrinetz:

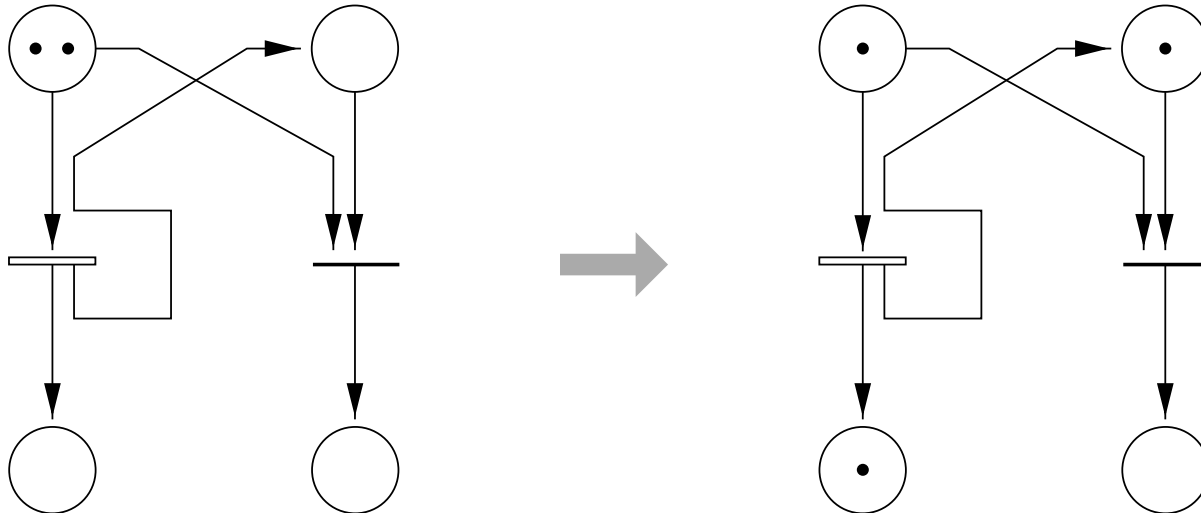
### ◆ System wird beschrieben durch:

- Transitionen und Stellen (Plätze)
- Kanten zw. Transitionen und Stellen bzw. zw. Stellen und Transitionen
- Marken in den Stellen

### ◆ Einfaches Beispiel:

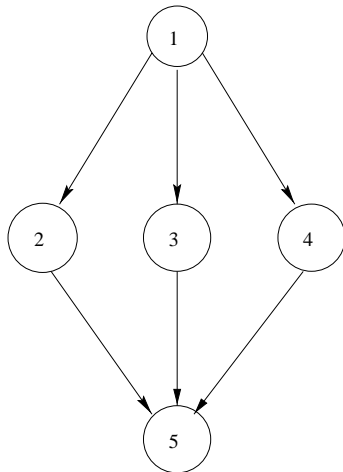


## ◆ Komplexeres Beispiel:

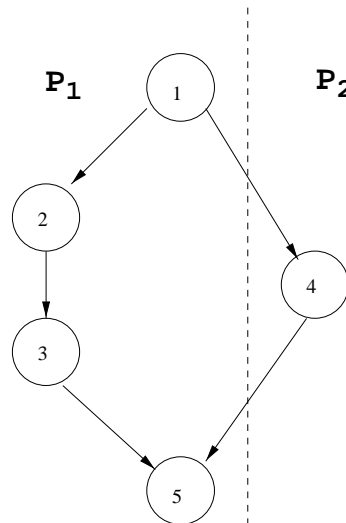


## ■ Präzedenzgraph:

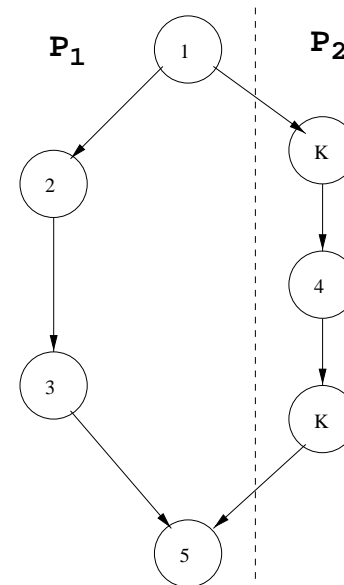
◆ Beispiel: 5 Teilaufgaben



3 Prozessoren



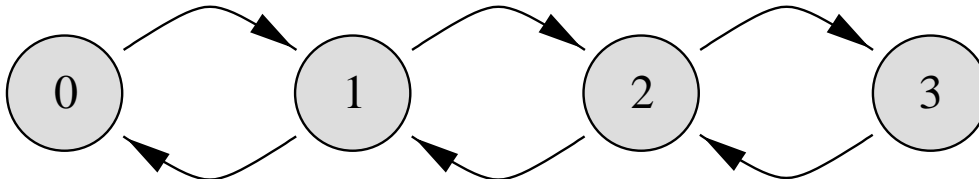
2 Prozessoren



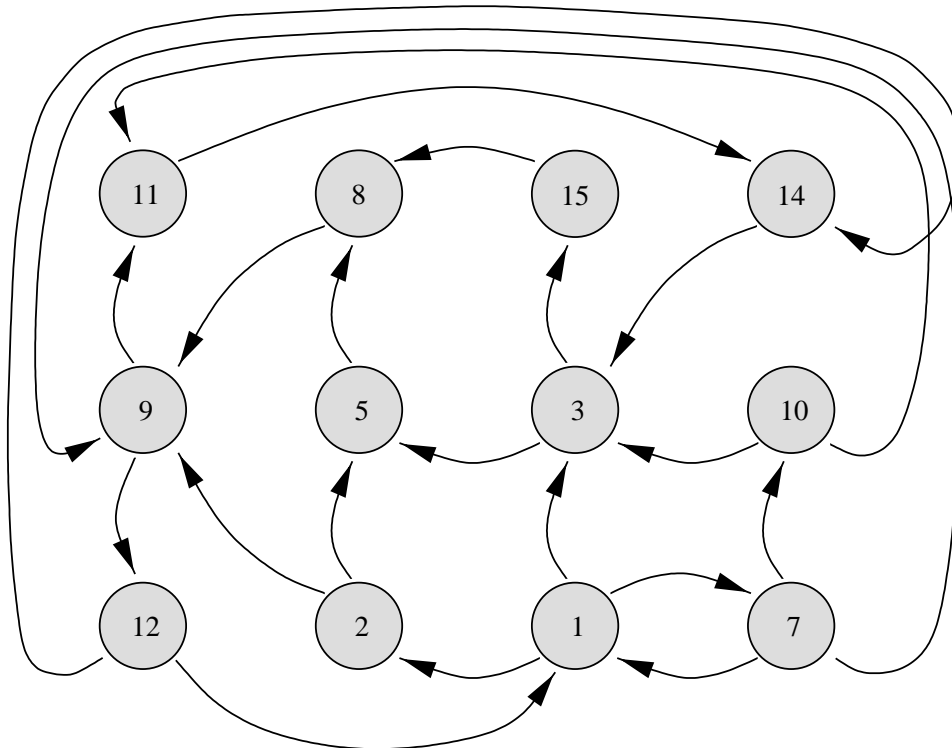
2 Prozessoren

## ■ Markov-Modell:

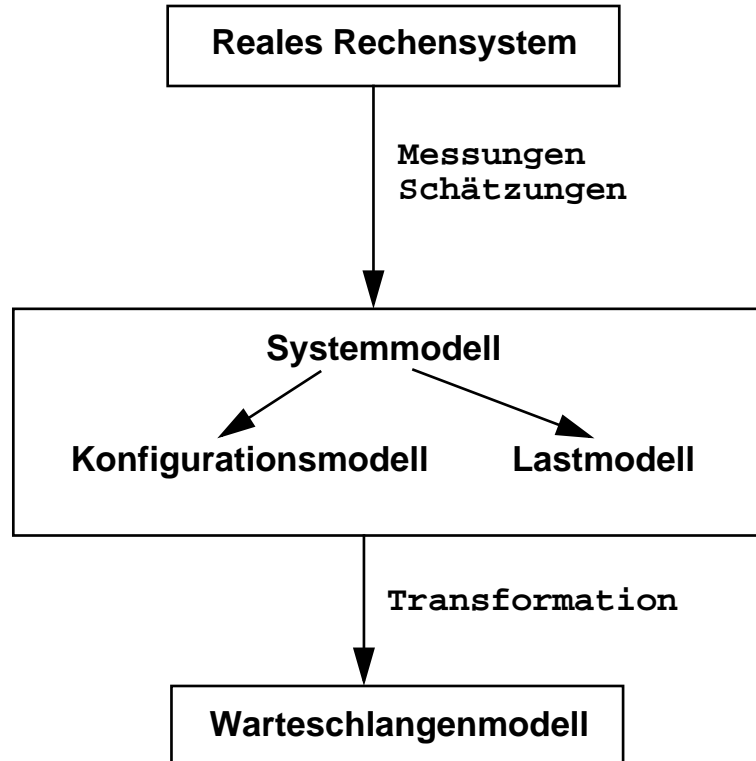
- ◆ System wird beschrieben durch die:
  - möglichen Zustände
  - möglichen Übergänge zwischen den Zuständen
  
- ◆ Einfaches Beispiel:



## ◆ Komplexeres Markovmodell:



# B.4 Modellierungsprozess



## B.5 Modelluntersuchungen

### ■ Simulation

- ◆ Nachspielen der Vorgänge mithilfe eines Programmes
  - Simulationsprogramme
  - Simulationsmodelle
- ◆ Herkömmliche Programmiersprachen
  - C, C++, JAVA, PASCAL, ...
- ◆ Spezielle Simulationssprachen
  - SIMULA, GPSS, ...
- ◆ Simulationswerkzeuge
  - SIMPLEX III, SIMPLE ++, ...
- ◆ Vorteil: System kann beliebig komplex sein
- ◆ Nachteil: Lange Rechenzeiten, Optimierungen umständlich

## ■ Analytische Methoden

### ◆ Darstellung des Zusammenhangs:

$$\begin{aligned}\text{Leistung} &= F(\text{Last, Software, Hardware}) \\ &= F(\text{Systemparameter})\end{aligned}$$

- Analytisches Modell

### ◆ Vorteil:

- Einfluss der Systemparameter leicht erkennbar
- Optimierung leicht durchführbar
- Kurze Rechenzeiten

### ◆ Nachteil:

- Für komplexe Systeme schwierig oder unmöglich exakte analytische Modelle zu erhalten
- Abhilfe: Approximative analytische Modelle

## ■ Forderungen an das Modell:

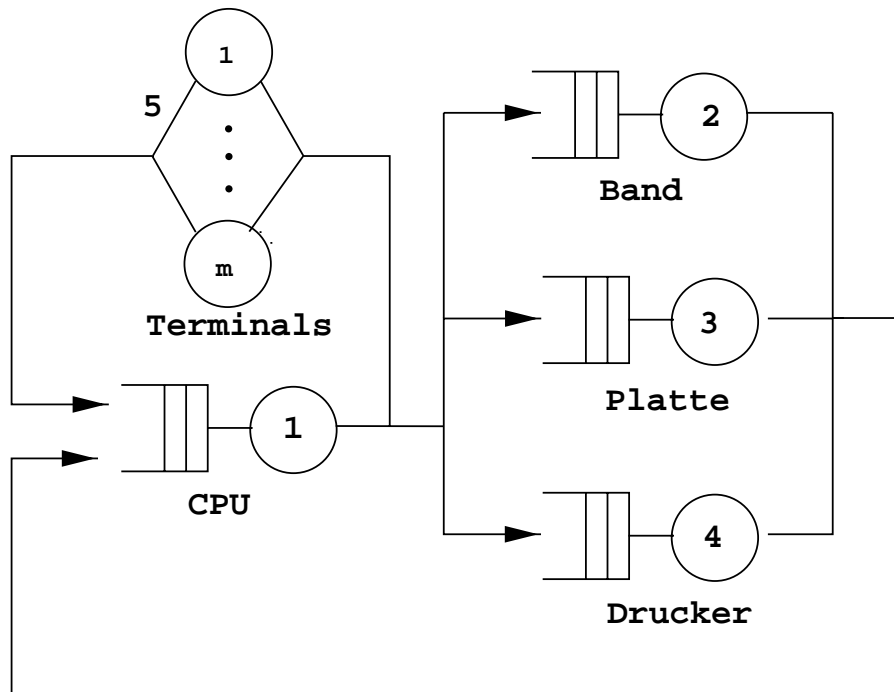
- ◆ Modellerstellung und Manipulation des Modells soll einfacher, billiger und schneller sein als Experiment am realen System
  
- ◆ Modell soll abbildungstreu sein, d.h. es soll alle relevanten Eigenschaften des realen Systems modellieren

## ■ Prinzipielle Vorgehensweise

- ◆ Rechensystem wird modelliert mithilfe von Warteschlangenmodellen (Oder anderen geeigneten Modelltypen)
  
- ◆ Annahmen oder Angaben über Systemparameter (Messungen, Schätzungen oder Angaben in Datenblättern)
  - Ankunftszeiten
  - Bearbeitungszeiten
  - Verteilung der Aufträge auf die Komponenten des Modells
  - meistens statistische Größen (Gegeben durch Mittelwert und Verteilung)
  
- ◆ Berechnung der Leistungsgrößen aus den Systemparametern mithilfe von Formeln und Algorithmen, die in der Vorlesung vorgestellt und teilweise auch hergeleitet werden

# B.6 Beispiele

## ■ Terminalsystem



## ◆ Systemparameter:

- CPU:
  - Zahl der Prozessoren: 3
  - Bedienzeit: 0.5 sec
  
- Band:
  - Bedienzeit: 5.0 sec
  
- Platte:
  - Bedienzeit: 1.0 sec
  
- Drucker:
  - Bedienzeit: 5.0 sec
  
- Terminals:
  - Denkzeit: 10 sec
  - Anzahl: 20

## ◆ Übergangswahrscheinlichkeiten

$$p_{12} = 0.15$$

$$p_{13} = 0.20$$

$$p_{14} = 0.15$$

$$p_{15} = 0.50$$

$$p_{21} = p_{31} = p_{41} = p_{51} = 1$$

◆ Bestimmung der Leistungsgrößen mithilfe des Leistungsbewertungstools **PEPSY** (**P**erformance **E**valuation and **P**rediction **S**ystem)

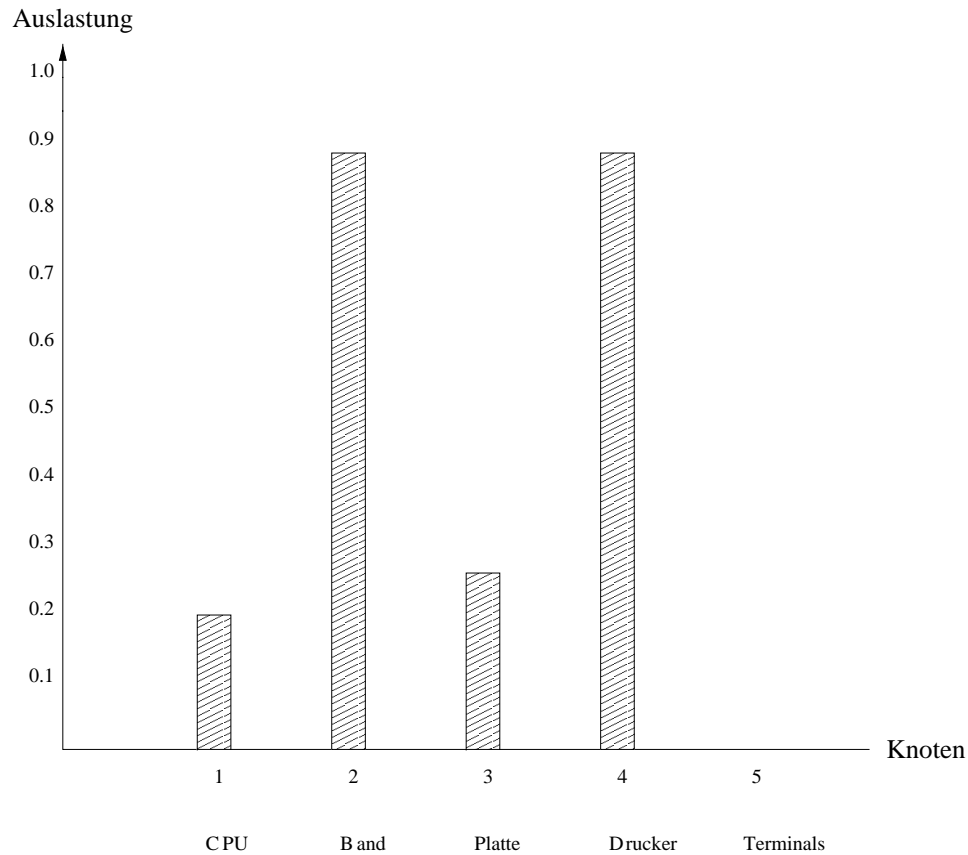
- Leistungsgrößen der einzelnen Komponenten (Knoten):

	Bedienzeit	Durchsatz	Auslastung	WS-Länge	Verweilzeit
CPU	0.500	1.147	<b>0.191</b>	0.005	0.504
Band	5.000	0.172	<b>0.860</b>	2.969	22.262
Platte	1.000	0.229	<b>0.229</b>	0.066	1.287
Drucker	5.000	0.172	<b>0.860</b>	2.969	22.262
Terminals	20.000	0.573	--	0.000	20.000

- Leistungsgrößen des Gesamtnetzes

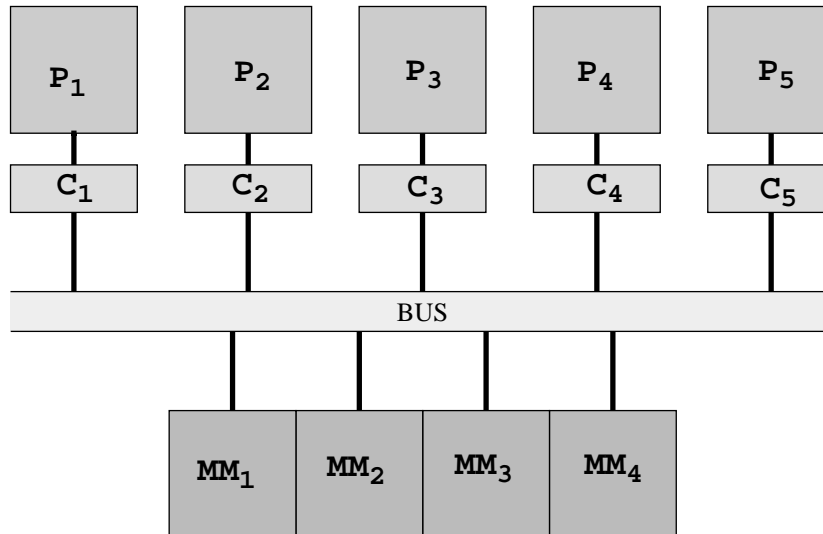
	Durchsatz	Verweilzeit
Netz	0.573	34.880

## ◆ Graphische Darstellung der Ergebnisse:



## ■ Multiprozessorsystem:

### ◆ Konfigurationsmodell:



$P_n$     Prozessor n

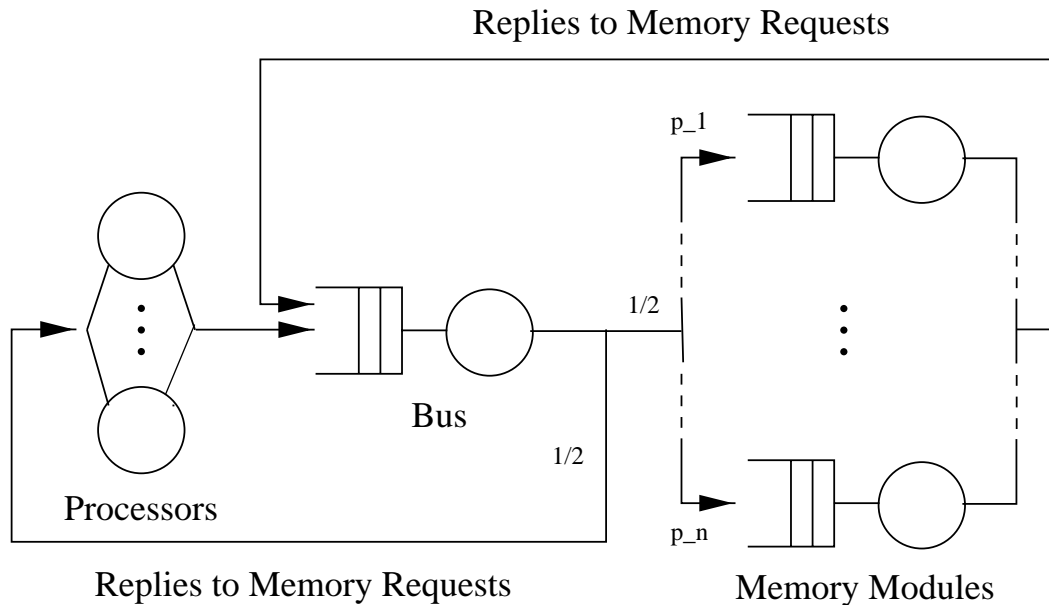
$C_n$     Cache n

$MM_n$     Memory Modul n

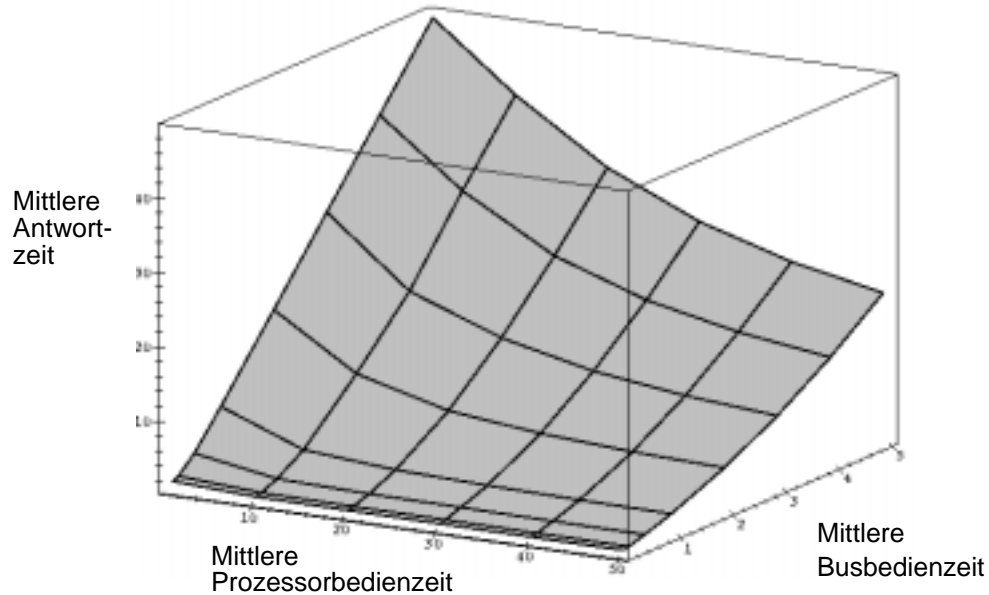
## ◆ Systemparameter:

- ▶ mittlere Busbedienzeit:  
Zeit, die der Bus im Mittel von einem Auftrag belegt ist
- ▶ mittlere Speicherbedienzeit:  
Zeit, die im Mittel für den Speicherzugriff benötigt wird
- ▶ mittlere Prozessorbedienzeit:  
mittlere Zeit zwischen zwei aufeinanderfolgende Speicherzugriffen
- ▶ Wahrscheinlichkeit  $p_n$  des Zugriffes auf den Speichermodul  $n$

◆ Warteschlangenmodell:



◆ Mittlere Antwortzeit für den Speicherzugriff eines Prozessors:



Mittlere Speicherbedienzeit = 1