Armin Wachter

Leistungsanalyse von IT-Systemen

Operationale Warteschlangentheorie und Performancetests

Dr. Armin Wachter
[redacted]
awachter@freenet.de

Independently Published

ISBN 9798336249552 (gebundene Ausgabe)
ISBN 9798340198846 (broschierte Ausgabe)

Bibliografische Informationen der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de> abrufbar.

Satz und Gestaltung: Dr. Armin Wachter unter Verwendung von LaTeX, Gnuplot und Microsoft PowerPoint.

Vorwort

Aus Nutzer- und Betreibersicht sind bei der Beurteilung der IT-Systemleistung drei Leistungsmerkmale besonders wichtig, nämlich *Antwortzeit*, *Durchsatz* und *Auslastung*. Die Antwortzeit beschreibt, wie schnell eine Anfrage vom System beantwortet wird. Der Durchsatz gibt an, wieviele Antworten pro Zeiteinheit vom System gesendet werden. Die Auslastung bemisst den Zeitanteil, in welchem das System aufgrund von Anfragen überhaupt beansprucht wird. Während Nutzer vornehmlich an kurzen Antwortzeiten interessiert sind, stehen für Betreiber in der Regel hohe Durchsätze und hohe Auslastungen bei gleichzeitig niedrigen Anschaffungs- und Betriebskosten im Vordergrund. Dies alles zu gewährleisten ist aus vielerlei Gründen nicht einfach, zum Beispiel weil kurze Antwortzeiten und hohe Auslastungen in einem natürlichen Spannungsverhältnis stehen.

Vor diesem Hintergrund beschäftigt sich die Leistungsanalyse von IT-Systemen hauptsächlich mit Fragen der folgenden Art:

- Durch welche Faktoren wird die Systemleistung beeinflusst?
- Wie hängt die Systemleistung konkret von diesen Faktoren ab?
- Wie lässt sich die Systemleistung prognostizieren und optimieren?

Die Leistungsanalyse geht somit über die reine Leistungsmessung im laufenden Betrieb *(Systemmonitoring)* oder im Rahmen von Experimenten *(Performancetests)* hinaus, indem sie vor allem die leistungsspezifischen Kausalzusammenhänge zu ergründen sucht. Diese Zusammenhänge wiederum sind notwendig, um die Systemleistung für Nicht-Messzeiträume bestimmen zu können.

Am Anfang einer jeden IT-systemischen Leistungsanalyse steht die *Systemmodellierung*. Hierbei wird das betreffende System in ein abstraktes und vereinfachendes Abbild seiner selbst, also in ein *Modell* überführt, wobei von allen Systemdetails abgesehen wird, die auf der angestrebten Betrachtungsebene für die Systemleistung irrelevant sind. Die Güte oder *Validität* des Modells bemisst sich danach, inwiefern es in der Lage ist, das Leistungsverhalten des Systems zu reproduzieren. Grundlage hierfür sind dedizierte Experimente am System, deren Rahmenbedingungen als *Modellinput* dienen und deren Messresultate mit dem *Modelloutput* verglichen werden. Mit zunehmender Zahl von erfolgreichen Modell-Experiment-Vergleichen wächst das Vertrauen, die relevanten systemischen Leistungsaspekte im Modell adäquat berücksichtigt zu haben. Mit einem auf diese Weise validierten Modell steht

am Ende ein mächtiges Werkzeug zur Verfügung, mit dessen Hilfe sich das systemische Leistungsverhalten unter den verschiedensten Rahmenbedingungen vorhersagen lässt.

Die prominentesten Vertreter von IT-Systemmodellen sind *Warteschlangenmodelle*. In ihnen wird ein IT-System als ein Netzwerk von *Stationen* dargestellt, die von einzelnen *Jobs* in unterschiedlicher Weise durchlaufen werden. Jede Station umfasst einen *Wartebereich*, in welchem Jobs auf Verarbeitung warten, und einen *Bedienbereich* mit ein oder mehreren *Prozessoren*, wo die eigentliche Jobverarbeitung stattfindet (siehe Abbildung 1). Wie sich zeigt, besitzen Warteschlangenmodelle

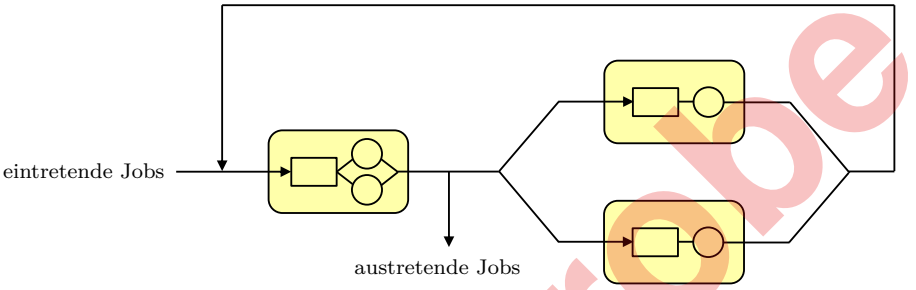


Abbildung 1. Netzwerk eines Warteschlangenmodells. In diesem Beispiel umfasst das Netzwerk einen Jobeingangs- und einen Jobausgangskanal, eine Zweiprozessorstation und zwei parallel geschaltete Einprozessorstationen. Die Rechtecke repräsentieren Wartebereiche, und die Kreise stehen für Prozessoren. Die Pfeile zeigen die Laufrichtung der Jobs durch das Netzwerk an. Ein Netzwerk heißt *geschlossen*, wenn in ihm eine feste Zahl von Jobs zirkuliert. Ansonsten ist es *offen* (so wie hier). Geschlossene Netzwerke eignen sich zur Modellierung von Batchsystemen oder von Systemen mit einer festen Zahl von aktiven Nutzern. Bei allen anderen Systemen sind offene oder gemischte Netzwerke zu bevorzugen.

einen geeigneten Abstraktionsgrad, um die Leistung von IT-Systemen in Form von Antwortzeiten, Durchsätzen, Auslastungen und weiteren „makroskopischen“ Leistungsmerkmalen effizient studieren zu können. Warteschlangenmodelle lassen sich hauptsächlich in drei Kategorien unterteilen:

(1) Stochastische Warteschlangenmodelle. Hierbei werden die meisten Modellfreiheitsgrade als *Zufallsvariablen* definiert. Der modellhafte Netzwerkbetrieb besteht deshalb im wesentlichen aus *stochastischen Prozessen* (also aus zeitlich geordneten Folgen von Zufallsvariablen), die sich unter sehr speziellen Bedingungen in mathematisch geschlossener Form ausarbeiten lassen. Infolgedessen sind natürlich auch die resultierenden Leistungsmerkmale größtenteils stochastischer Natur. Die Wurzeln der stochastischen Warteschlangentheorie reichen zurück bis zum Anfang des zwanzigsten Jahrhunderts, als sich der dänische Mathematiker Agner Krarup Erlang mit Wartezeitproblemen bei der zentralen Vermittlung von Telefongesprächen beschäftigte. Spätestens seit den 1960er Jahren gehört diese umfangreiche, mathematisch anspruchsvolle und auch heute noch nicht abgeschlossene Theorie zu den Standardwerkzeugen der IT-systemischen Leistungsanalyse.

(2) Simulative Warteschlangenmodelle. Ihnen liegt die Idee zugrunde, konkrete Ausprägungen der modellhaften Netzwerkdynamik im Detail nachzustellen. In der besonderen Form der *diskreten Ereignissimulation* wird die Reise eines jeden Jobs durch das Netzwerk mit Hilfe eines Computerprogramms und auf der Basis eines wohldefinierten (pseudozufälligen oder deterministischen) Regelwerkes akkurat verfolgt und an den Ereigniszeitpunkten, zum Beispiel wenn der Job eine Station betritt oder verlässt, protokolliert. Die Auswertung der protokollierten Daten geschieht in ähnlicher Weise wie bei Experimenten an realen Systemen. Ein Vorteil des simulativen Ansatzes ist, dass er im Prinzip keinen Einschränkungen bezüglich der Implementierung von speziellen Netzwerkfunktionalitäten unterworfen ist. Deshalb lassen sich hiermit auch Systeme analysieren, die anderen Analysemethoden nicht zugänglich sind. Dem steht allerdings der Nachteil gegenüber, dass die Entwicklung und Durchführung von Simulationen oftmals sehr aufwändig ist.

(3) Operationale Warteschlangenmodelle. Wie stochastische Modelle besitzen sie einen geschlossenen mathematischen Formalismus. Allerdings werden die modellhaften Voraussetzungen und Leistungsgrößen nicht stochastisch sondern *operational* definiert, mit der Bedeutung, dass sie auf einen festen Zeitraum bezogen sind und in genau diesem Zeitraum am realen System geprüft bzw. gemessen werden können. Entscheidend hierbei ist nicht, dass die Prüfungen und Messungen tatsächlich stattfinden, sondern dass sie *prinzipiell* möglich sind. Um dies zu verdeutlichen, betrachte man einen Webserver, dessen modellhaftes Netzwerk aus einer einzigen Station mit einem Jobeintritts- und einem Jobaustrittskanal besteht. Im Rahmen der stochastischen Theorie bestünde eine typische Modellvoraussetzung darin, dass die Zwischeneintrittszeiten der Jobs unabhängig und identisch exponentialverteilt sind. Diese Voraussetzung lässt sich schon allein aufgrund der Bedeutung des Begriffes „Zufallsvariable“ weder exakt noch ungefähr am Webserver prüfen; sie ist daher nicht-operational. Im Rahmen der operationalen Theorie könnte man dagegen annehmen, dass in einem bestimmten Zeitraum die stationsseitigen Jobeintritts- und Jobaustrittszahlen übereinstimmen. Diese Voraussetzung ist in der Tat operational, weil man sie am Webserver einfach durch Zählung der eingehenden Anfragen und gesendeten Antworten im besagten Zeitraum exakt prüfen kann.

Die operationale Warteschlangentheorie wurde von Peter J. Denning, Jeffrey P. Buzen und anderen Informatikern Mitte der 1970er Jahre eingeführt und kann als eine Reaktion auf folgende Beobachtungen im praktischen Umgang mit der stochastischen Theorie aufgefasst werden:

- Stochastische Voraussetzungen sind ihrem Wesen nach grundsätzlich nicht experimentell verifizierbar (wie soeben diskutiert). Infolgedessen kann ein stochastisches Warteschlangenmodell streng genommen nicht validiert werden.
- Stochastische Modelle sind in der Lage, das Leistungsverhalten vieler IT-Systeme mit erstaunlicher Genauigkeit vorherzusagen, und zwar auch dann, wenn die Systeme einige Modellvoraussetzungen mit ziemlicher Sicherheit nicht erfüllen. Demnach scheinen stochastische Modelle über „versteckte Merkmale“ zu verfügen, die die eigentlichen Ursachen des Erfolges sind.

- Ein Systemanalyst, der mit konkreten praktischen Leistungsfragen konfrontiert ist, tut sich in der Regel schwer, diese in einem stochastischen Kontext zu sehen und zu bearbeiten.

Gegenüber der stochastischen Theorie zeichnet sich die operationale Theorie vor allem durch drei Eigenschaften aus:

- Die stochastische Theorie macht Aussagen über alle möglichen konkreten Ausprägungen der Netzwerkdynamik, die mit den gegebenen Voraussetzungen im Einklang stehen, und zwar in Form von Wahrscheinlichkeitsverteilungen, Erwartungswerten, Varianzen und ähnlichem.[1] In der operationalen Theorie können dagegen alle voraussetzungskonformen netzwerkdynamischen Ausprägungen als *operational identisch* betrachtet werden. Es genügt daher, sich gedanklich auf irgendeine dieser Ausprägungen zu beschränken und diese eine mit Hilfe von einfachen Mittelwerten über den definierten Messzeitraum zu beschreiben.
- Dieser fundamentale Unterschied bedeutet, dass die operationalen Leistungsgrößen einen anderen und auch geringeren Informationsgehalt besitzen als die stochastischen. Dafür bietet die operationale Theorie einen intuitiveren und mathematisch einfacheren Formalismus, der vor allem dem Praktiker entgegenkommt.
- Erstaunlicherweise ist die operationale Theorie trotzdem in der Lage, viele zentrale Gesetzmäßigkeiten der stochastischen Theorie zu reproduzieren, natürlich vorbehaltlich der damit verbundenen Interpretationsunterschiede. Beispiele hierfür sind das *Littlesche Gesetz*, die *Ankunftstheoreme* sowie die *Produktformlösungen für separable Netzwerke*.

Performancetests. Grundsätzlich ist man bei jeder Art der IT-systemischen Leistungsevaluierung immer auch auf Leistungsmessungen von realen IT-Systemen angewiesen. Hierzu kommen in der Regel *Performancetests* zum Einsatz, in denen jeweils ein bestimmtes System in offener oder geschlossener Form über einen längeren Zeitraum hinweg künstlich unter Last gesetzt wird und die diesbezüglichen Systemreaktionen (Antwortzeiten, Durchsätze, Auslastungen etc.) fortlaufend gemessen werden. Fokussiert man sich hierbei auf die messdatenanalytischen Aspekte, so sind vor allem folgende Punkte hervorzuheben:

- Jeder Performancetest ist zwangsläufig unkontrollierbaren und unvorhersehbaren Einflüssen ausgesetzt. Deshalb ist es angebracht, (i) den Test als ein *dynamisches Zufallsexperiment*, (ii) die mit dem Test verbundenen Messvorgänge als *stochastische Leistungsprozesse* (also als zeitlich geordnete Folgen von Zufallsvariablen) und (iii) die aus den Messvorgängen resultierenden Messreihen als konkrete *Realisierungen* der Prozesse aufzufassen.
- Demzufolge besteht die Hauptaufgabe der Messdatenanalyse darin, die interessierenden Verteilungsparameter (Erwartungswerte, Varianzen etc.) eines jeden Leistungsprozesses anhand der vorliegenden, in ein oder mehreren Testwiederho-

[1] Man kann sich eine konkrete Ausprägung der Netzwerkdynamik als eine Netzwerksimulation mit bestimmten Anfangsbedingungen und einer bestimmten Initialisierung des Pseudo-Zufallszahlengenerators vorstellen.

lungen generierten Prozessrealisierungen möglichst akkurat zu schätzen (*induktive* oder *inferenzielle Statistik*).

- Die hierfür zur Verfügung stehenden statistischen Analysemethoden adressieren unterschiedliche Interessenschwerpunkte und beruhen größtenteils auf der Strategie, das jeweilige Schätzproblem auf ein Standardproblem der *statistischen Schätztheorie* zurückzuführen (siehe Abbildung 2).

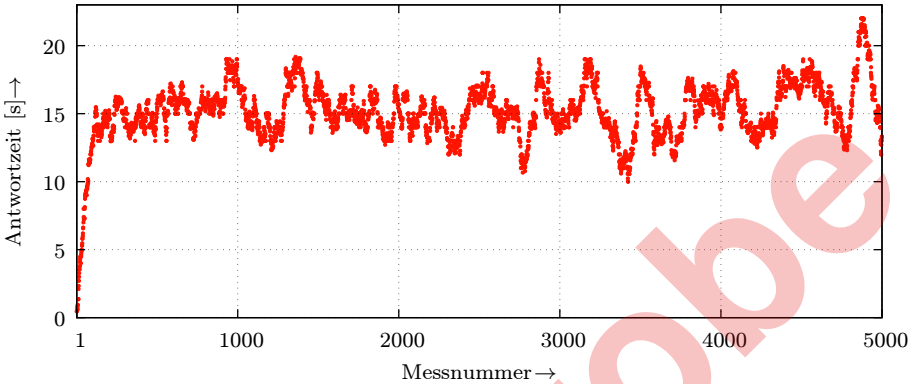


Abbildung 2. Gemessene Antwortzeiten eines Webservers im Rahmen eines geschlossenen Performancetests mit konstantem Lastprofil. Aus der stochastischen Perspektive entspricht die Messreihe einer konkreten Realisierung des abstrakten serverseitigen Antwortzeitprozesses. Offensichtlich oszillieren die Messwerte wellenförmig, und zwar dergestalt, dass auf einen kleinen (großen) Wert ein ähnlich kleiner (großer) Wert folgt. Dies deutet darauf hin, dass testzeitlich eng benachbarte Prozessvariablen positiv korreliert sind. Ferner erkennt man anhand der linken Messwerte, dass der Prozess eine gewisse Zeit benötigt, um sich „einzuschwingen“. Möchte man den stationären (langfristigen) Prozesserwartungswert allein auf der Basis dieser einen Prozessrealisierung bestimmen, so kann man hierfür die statistische Schätztheorie nicht direkt in Anschlag bringen, weil die einzelnen Prozessvariablen (i) nicht unabhängig voneinander und (ii) in der Einschwingphase nicht identisch verteilt sind. Deshalb böte es sich hier an, die Messreihe dergestalt zu transformieren, dass sie als Realisierung eines anderen Prozesses aufgefasst werden kann, der seinerseits in guter Näherung aus einer Folge von unabhängigen und identisch verteilten Zufallsvariablen besteht und darüber hinaus den gesuchten Erwartungswert besitzt. Notwendige Voraussetzung hierfür ist allerdings, dass der ursprüngliche Antwortzeitprozess *ergodisch* ist und somit seinen momentanen Zustand hinreichend schnell vergisst.

Buchinhalte. Das vorliegende Buch behandelt die Leistungsanalyse von IT-Systemen mittels der operationalen Warteschlangentheorie und Performancetests. Es richtet sich an Systemarchitekten und Systemanalysten, die mit der Planung, Messung, Bewertung oder Prognose des Leistungsverhaltens von IT-Systemen befasst sind. Eine weitere Zielgruppe sind Dozenten und Studenten der theoretischen Informatik. Für sie liefert das Buch eine gedankliche Basis, auf die viele Konzepte und Resultate der stochastischen Warteschlangentheorie heruntergebrochen werden können, um auf diese Weise ein intuitiveres Verständnis der Materie zu generieren. Schließlich können auch all jene von dem Buch profitieren, die sich ganz allgemein über praxisnahe Konzepte der IT-systemischen Leistungsanalyse informieren möchten.

Das Buch ist in vier Kapitel und zwei Anhänge unterteilt. Das erste Kapitel beschäftigt sich mit den Grundzügen der operationalen Warteschlangentheorie. Hierzu gehören die Beschreibung des operationalen Basismodells, die Entwicklung der wichtigsten Leistungsgrößen und allgemeinsten Gesetze sowie die konzeptionelle Erweiterung der Theorie in Form der sogenannten *homogenen Netzwerke*. Sie zeichnen sich dadurch aus, dass das Verhalten einer jeden Station ausschließlich von ihrer eigenen momentanen *Besetzung* mit Jobs abhängt. In diesem Zusammenhang werden auch die Begriffe *Zustand*, *Zustandsübergang* und *Zustandsverteilung* (in einem nicht-stochastischen Sinne) eingeführt, die für die exakte Bestimmung des netzwerkseitigen Leistungsverhaltens im dritten Kapitel von zentraler Bedeutung sind.

Im zweiten Kapitel stellen wir die wichtigsten Analysetechniken für geschlossene homogene Netzwerke vor, bei denen auf die Kenntnis der Zustandsverteilung verzichtet werden kann. Dementsprechend handelt es sich hierbei vornehmlich um Eingrenzungs- und Approximationstechniken. Ihr Vorteil besteht darin, dass sie schnell und mit leichter Hand zum Einsatz gebracht werden können und trotzdem Vorhersagen liefern, die in vielen Anwendungsfällen hinreichend genau sind.

Das dritte Kapitel handelt von der exakten Leistungsberechnung von offenen und geschlossenen homogenen Netzwerken. Dies läuft im wesentlichen auf das Lösen von linearen Gleichungssystemen für die Zustandsverteilung hinaus. Das besondere an den Lösungen ist, dass sie sich aus den Lösungen der einzelnen, in Isolation betrachteten Stationen multiplikativ zusammensetzen *(Produktformlösungen)*. Weil die Auswertung der Lösung im geschlossenen Fall recht komplex ist, beschäftigen sich gleich mehrere Abschnitte damit.

Das vierte Kapitel ist der Konzeption und statistischen Analyse von Performancetests gewidmet, wobei wir uns auf solche Tests beschränken, bei denen die quantitative Bestimmung von IT-systemischen Leistungsgrößen – etwa im Kontext des direkten Leistungsvergleiches von IT-Systemen oder der Parametrisierung/Validierung von IT-Systemmodellen – im Vordergrund steht. Das vorrangige Ziel des Kapitels ist die Erarbeitung von einfachen und praktikablen Methoden zur statistischen Schätzung von bestimmten Verteilungsparametern der mit einem Performancetest verbundenen Leistungsprozesse. Auf dem Weg dorthin wird es sich als hilfreich erweisen, zunächst den stochastischen Charakter von Performancetests herauszuarbeiten und anschließend einige Bereiche der statistischen Schätztheorie zu diskutieren.

Im ersten Anhang werden schließlich die wahrscheinlichkeitstheoretischen Grundlagen bereitgestellt, die im vierten Kapitel benötigt werden.

Das Buch ist als Lehr- und Übungsbuch konzipiert. Wichtige Voraussetzungen und Beziehungen sind in Definitions- und Satzkästen zusammengefasst, um so dem Leser ein strukturiertes Lernen und schnelles Nachschlagen zu ermöglichen. Desweiteren befindet sich am Ende eines jeden Kapitels eine Zusammenfassung, gefolgt von vielen Aufgaben mit Lösungen, mittels derer das Verständnis des behandelten Stoffes überprüft werden kann. Einige der insgesamt 130 Aufgaben gehen auch über den Lehrstoff hinaus. Die ersten drei Kapitel bewegen sich mathematisch auf

niedrigem Niveau und setzen lediglich Kenntnisse in linearer Algebra und Kombinatorik voraus. Für das Verständnis des vierten Kapitels und des ersten Anhangs sind zusätzlich Kenntnisse der Differential- und Integralrechnung notwendig.

Dieses Buch erhebt keinen Anspruch auf Vollständigkeit. Stattdessen wurden seine Inhalte so ausgewählt, dass sie einerseits die grundlegenden Konzepte der operationalen Warteschlangentheorie sowie der statistischen Analyse von Performancetests enthalten und andererseits den Bedürfnissen des Praktikers – die Anwendung dieser Theorien auf konkrete IT-systemische Leistungsfragen – gerecht werden. Auf der Grundlage des hier präsentierten Stoffes sollte der Leser in der Lage sein, sich weitere Themen selbstständig zu erarbeiten. Auch hierzu sind die kommentierten Literaturvorschläge im zweiten Anhang hilfreich.

Köln im August 2024 *Armin Wachter*

Inhaltsverzeichnis

Symbolverzeichnis

Warteschlangentheoretische Symbole

A	Zahl der Eintritte *(entries)*, Zahl der Ankünfte *(arrivals)*
B	Betriebszeit *(busy time)*, Nicht-Leerzeit
C	Zahl der Austritte *(completions, departures)*
$D, d; \hat{D}$	Bedienaufwand *(service demand)*; Prozessoraufwand *(processor demand)*
F	Akkumulierte Verweilzeit *(accumulated residence time)*, Stationsfunktion *(device function)*
$G; g$	Normierungskonstante *(normalization constant)*; Normierungskonstante von Zwischensystemen
M	Zahl der Stationen *(devices)*
m	Zahl der Prozessoren *(processors, service units)*, Zahl der Stationen von Zwischensystemen
N	Gesamtjobzahl *(total job number, multiprogramming level)*, Besetzungskapazität *(queue capacity)*
n	Besetzungszahl *(occupation number, current queue length)*, Gesamtjobzahl von Zwischensystemen
$\boldsymbol{n}$	Systemzustand *(system state)*
$p(n); p(\boldsymbol{n})$	Besetzungsverteilung *(queue length distribution)*; systemische Zustandsverteilung *(system state distribution)*
Q	Jobzahl *(queue length)*
$q; \boldsymbol{Q}$	Routingfrequenz *(routing frequency)*; Routingmatrix
R	Antwortzeit *(response time)*, Verweilzeit *(residence time)*
$S; \hat{S}$	Bedienzeit *(service time)*; Prozessorzeit *(processor time)*
T	Messzeitraum, Beobachtungszeitraum *(observation period)*
$U; \hat{U}$	Auslastung *(utilization)*; effektive Auslastung
V	Besuchsverhältnis *(visit ratio)*

W	Wartezeit *(waiting time)*
X	Durchsatz*(throughput)*, Austrittsrate *(completion rate)*
Z	Denkzeit *(think time)*, Verzögerung *(delay)*
γ	Abweisungsrate *(rejection rate)*
λ	Eintrittsrate *(entering rate)*, Ankunftsrate *(arrival rate)*
$\mu; \hat{\mu}$	Bedienrate *(service rate)*; Prozessorrate *(processor rate)*
$\Omega; \omega$	Zustandsraum *(state space)*; Zustandsraum von Zwischensystemen

Allgemeine mathematische Symbole

$\mathbb{N}, \mathbb{N}_0$	Menge der natürlichen Zahlen exklusive/inklusive der Null
$\mathbb{R}; \mathbb{R}^+, \mathbb{R}_0^+$	Menge der reellen Zahlen; Menge der positiven reellen Zahlen exklusive/inklusive der Null
$\{x \mid A(x)\}$, $\{x : A(x)\}$	Beschreibende Darstellung einer Menge: Menge aller x, für die die Aussage $A(x)$ zutrifft
$[a : b]$	Menge der natürlichen, ganzen oder reellen Zahlen von a bis b
$\#A$	Anzahl der Elemente bzw. Mächtigkeit der Menge A
$A \cup B$	Vereinigungsmenge der Mengen A und B
$A \cap B$	Schnittmenge der Mengen A und B
$x \in A$	x ist Element der Menge A
$x \notin A$	x ist nicht Element der Menge A
$\lfloor x \rfloor, \lceil x \rceil$	Ganzzahlige Abrundung/Aufrundung von x
$x = \text{const}$	x ist eine Konstante
$x \approx y$	x ist ungefähr gleich y
$x \lesssim y$	x ist ungefähr gleich oder kleiner als y
$x \gtrsim y$	x ist ungefähr gleich oder größer als y
$x \ll y$	x ist sehr viel kleiner als y
$x \gg y$	x ist sehr viel größer als y
$x \propto y$	x ist proportional zu y
$\forall x$	für alle x
$x \vee y$	x oder y
$x \wedge y$	x und y

$\sum_{i=1}^{n} x_i$	Summe der x_i von $i = 1$ bis $i = n$
$\prod_{i=1}^{n} x_i$	Produkt der x_i von $i = 1$ bis $i = n$
$\boldsymbol{n}$	$(n_1, n_2, \ldots)$, sprich: „Vektor n“
$n!$	$1 \cdot 2 \cdot \ldots \cdot n$, Fakultät von n, sprich: „n Fakultät“
$\binom{n}{k}$	$\frac{n!}{(n-k)!k!}$, Binomialkoeffizient, sprich: „n über k“
$\lim_{x \to a} f(x)$	Grenzwert der Funktion $f(x)$, wenn x gegen a strebt, sprich: „Limes von $f(x)$ für x gegen a“
$f(x) \overset{x \to a}{\longrightarrow}$	Andere Schreibweise für $\lim_{x \to a} f(x)$
$f(x) = \left\{ \begin{matrix} g(x) \\ h(x) \end{matrix} \right. \left. \begin{matrix} , \; x \leq a \\ , \; \text{sonst} \end{matrix} \right\}$	Abschnittsweise definierte Funktion: $f(x)$ ist gleich $g(x)$ für alle $x \leq a$, $f(x)$ ist gleich $h(x)$ für alle sonstigen x
$A \Rightarrow B$	Implikation mit zwei kontextabhängigen Bedeutungen: (1) Es gilt Aussage A, und daraus folgt Aussage B (2) Aus Aussage A folgt Aussage B
$A \Leftarrow B$	Implikation mit zwei kontextabhängigen Bedeutungen: (1) Es gilt Aussage B, und daraus folgt Aussage A (2) Aus Aussage B folgt Aussage A
$A \Leftrightarrow B$	Äquivalenz mit zwei kontextabhängigen Bedeutungen: (1) Es gelten die Aussagen A und B, und beide folgen auseinander (2) Aus Aussage A folgt Aussage B und umgekehrt
♦	Beweis- oder Begründungsende

Wahrscheinlichkeitstheoretische Symbole

$\omega; \Omega$	Ergebnis; Ergebnisraum
E	Ereignis, Teilmenge des Ergebnisraumes Ω
$\mathcal{I}\{E\}$	Indikatorvariable des Ereignisses E
$E \perp F$	Die Ereignisse E und F sind unabhängig voneinander
$\mathbb{P}\{E\}$	Wahrscheinlichkeit des Ereignisses E
$\mathbb{P}\{E \cup F\}$	Wahrscheinlichkeit für die Vereinigungsmenge der Ereignisse E und F
$\mathbb{P}\{E \cap F\}$	Wahrscheinlichkeit für die Schnittmenge der Ereignisse E und F

$\mathbb{P}\{E\|F\}$	Wahrscheinlichkeit des Ereignisses E unter der Bedingung von Ereignis F
$X; x$	Zufallsvariable; Realisierung der Zufallsvariablen
$X \perp Y$	Die Zufallsvariablen X und Y sind unabhängig voneinander
$p_X(x)$	Wahrscheinlichkeitsfunktion der diskreten Zufallsvariablen X
$f_X(x)$	Wahrscheinlichkeitsfunktion/Wahrscheinlichkeitsdichte der stetigen Zufallsvariablen X
$F_X(x)$	Verteilungsfunktion der Zufallsvariablen X
$p_{X,Y}(x,y)$	Gemeinsame Wahrscheinlichkeitsfunktion der diskreten Zufallsvariablen X und Y
$f_{X,Y}(x,y)$	Gemeinsame Wahrscheinlichkeitsfunktion der stetigen Zufallsvariablen X und Y
$F_{X,Y}(x,y)$	Gemeinsame Verteilungsfunktion der Zufallsvariablen X und Y
$p_{X\|Y}(x\|y)$	Bedingte Wahrscheinlichkeitsfunktion der diskreten Zufallsvariablen X unter der Bedingung $\{Y = y\}$
$f_{X\|Y}(x\|y)$	Bedingte Wahrscheinlichkeitsfunktion der stetigen Zufallsvariablen X unter der Bedingung $\{Y = y\}$
$F_{X\|Y}(x\|y)$	Bedingte Verteilungsfunktion der Zufallsvariablen X unter der Bedingung $\{Y = y\}$
$\mathbb{E}[X]$	Erwartungswert der Zufallsvariablen X
$\mathbb{E}[X\|Y = y]$	Erwartungswert der Zufallsvariablen X unter der Bedingung $\{Y = y\}$
$\mathbb{V}[X]$	Varianz der Zufallsvariablen X
$\mathbb{V}[X\|Y = y]$	Varianz der Zufallsvariablen X unter der Bedingung $\{Y = y\}$
$\mathrm{Cov}[X,Y]$	Kovarianz der Zufallsvariablen X und Y
$\mathrm{Corr}[X,Y]$	Korrelation der Zufallsvariablen X und Y
$X \sim \ldots$	Die Zufallsvariable X genügt der Verteilung ...
$\mathrm{U}(a,b)$	Diskrete oder stetige Gleichverteilung auf dem Intervall $[a:b]$
$\mathrm{Geo}(p)$	Diskrete geometrische Verteilung mit der Erfolgswahrscheinlichkeit p
$\mathrm{Bin}(n,p)$	Diskrete Binomialverteilung mit n Wiederholungen und der Erfolgswahrscheinlichkeit p
$\mathrm{Exp}(\lambda)$	Stetige Exponentialverteilung mit der Ereignisrate λ
$\mathrm{Erl}(\lambda,n)$	Stetige Erlang-Verteilung für die Summe von n wechselseitig unabhängigen und $\mathrm{Exp}(\lambda)$-verteilten Zufallsvariablen
$\mathrm{N}(\mu,\sigma^2); z_q$	Stetige Normalverteilung mit Erwartungswert μ und Varianz σ^2; Quantile der Standard-Normalverteilung $\mathrm{N}(0,1)$

$\chi^2(n); x_{n,q}$	Stetige χ^2-Verteilung mit n Freiheitsgraden; Quantile der $\chi^2(n)$-Verteilung
$\mathrm{St}(n); t_{n,q}$	Stetige Student-Verteilung mit n Freiheitsgraden; Quantile der $\mathrm{St}(n)$-Verteilung
$\{Y(t), t \in I\}$	Stochastischer Prozess mit der Indexmenge I
$\gamma(s)$	Autokovarianzfunktion eines stationären Prozesses
$\rho(s)$	Autokorrelationsfunktion eines stationären Prozesses

Statistische Symbole

n	Stichprobenumfang
$\{X_1, \ldots, X_n\}$; $\{x_1, \ldots, x_n\}$	Stichprobe der Zufallsvariablen X; Realisierung der Stichprobe
$\mathcal{E}_\theta(n); e_\theta(n)$	Punktschätzer des Verteilungsparameters θ; Realisierung des Punktschätzers
$\bar{X}(n); \bar{x}(n)$	Stichprobenmittel der Zufallsvariablen X; Realisierung des Stichprobenmittels
$\tilde{S}^2(n); \tilde{s}^2(n)$	Stichprobenvarianz einer Zufallsvariablen mit bekanntem Erwartungswert; Realisierung der Stichprobenvarianz
$S^2(n); s^2(n)$	Stichprobenvarianz einer Zufallsvariablen mit unbekanntem Erwartungswert; Realisierung der Stichprobenvarianz
$\tilde{C}_{XY}(n)$; $\tilde{c}_{XY}(n)$	Stichprobenkovarianz der Zufallsvariablen X und Y mit bekannten Erwartungswerten; Realisierung der Stichprobenkovarianz
$C_{XY}(n)$; $c_{XY}(n)$	Stichprobenkovarianz der Zufallsvariablen X und Y mit unbekannten Erwartungswerten; Realisierung der Stichprobenkovarianz
$K_\theta(n,p)$; $k_\theta(n,p)$	Konfidenzintervall(schätzer) des Verteilungsparameters θ auf dem Konfidenzniveau p; Realisierung des Konfidenzintervalls
$\mathcal{R}; r$	Prüfregel zur statistischen Signifikanz; Realisierung der Prüfregel
$\alpha; \beta$	Irrtumswahrscheinlichkeit/Signifikanzniveau einer Prüfregel; Ignoranzwahrscheinlichkeit einer Prüfregel

Kapitel 1

Grundlagen der operationalen Warteschlangentheorie

Warteschlangenmodelle basieren auf der Idee, dass sich jedes IT-System auf einer hohen Abstraktionsebene durch ein Netzwerk von Bedienstationen (ähnlich den Kassen in Supermärkten oder den Check-In-Schaltern in Flughäfen) beschreiben lässt. In dieser Vorstellung gleicht der IT-Systembetrieb einem Strom von Jobs, die in das Netzwerk eintreten, einzelne Stationen besuchen, in diesen Stationen verarbeitet werden und schließlich das Netzwerk wieder verlassen. Je nachdem, wie ein solches Netzwerk definiert und hinsichtlich seiner Leistungseigenschaften analysiert wird, unterscheidet man vor allem stochastische, simulative und operationale Warteschlangenmodelle, von denen der operationale Typ der Hauptgegenstand der ersten drei Buchkapitel ist.

Dieses Kapitel beschäftigt sich mit den grundlegenden Konzepten und Zusammenhängen der operationalen Warteschlangentheorie. In den ersten beiden Abschnitten erläutern wir den operationalen Ansatz und stellen das allgemeine operationale Warteschlangennetzwerk zusammen mit seinen primären Voraussetzungen und Leistungsgrößen vor. In den Abschnitten 1.3 bis 1.8 werden die wichtigsten Gesetzmäßigkeiten zwischen den Leistungsgrößen entwickelt. Diese Gesetze besitzen einen großen Gültigkeitsbereich und können daher in nahezu allen IT-systemischen Leistungsanalysen zum Einsatz gebracht werden. In Abschnitt 1.9 führen wir als operationale Erweiterung das Konzept der homogenen Netzwerke ein, welches in den Kapiteln 2 und 3 die zentrale Rolle spielen wird. Der letzte Abschnitt 1.10 beschäftigt sich mit Aspekten, die bei der operationalen Modellierung von IT-Systemen – jenseits der Etablierung von Leistungsgrößen und Leistungsgesetzen – von prinzipieller Bedeutung sind. Insbesondere werden dort die drei Phasen des Modellierungszyklus diskutiert, nämlich die Modellentwicklung, die Modellvorhersage und die Modellverifikation.

1.1 Operationaler Ansatz

Im Rahmen der operationalen Warteschlangentheorie wird das Leistungsverhalten von IT-Systemen mit Hilfe von *operationalen Warteschlangenmodellen* evaluiert, die sich ihrerseits wie folgt beschreiben lassen:

Definition 1.1: Operationales Warteschlangenmodell
Ein operationales Warteschlangenmodell ist eine abstrakte und vereinfachende Darstellung eines IT-Systems in Form eines Netzwerkes von Stationen *(Warteschlangennetzwerk)*. Hieran sind bestimmte Voraussetzungen und Leistungsgrößen geknüpft, die sich vor allem dadurch auszeichnen, dass sie alle auf einen festen Zeitraum bezogen sind und in diesem Zeitraum prinzipiell am IT-System geprüft bzw. gemessen werden können *(operationales Prinzip)*.

Bei dieser Definition sind folgende Punkte zu beachten:

- Die Beschreibung des Netzwerkes erfolgt rein *phänomenologisch*. Damit ist gemeint, dass sich die netzwerkseitigen Voraussetzungen nicht auf intrinsisch-funktionale Eigenschaften beziehen, die für das Leistungsverhalten des Netzwerkes ursächlich sind, sondern auf bestimmte Eigenschaften der Daten, die am Netzwerk fiktiv gemessen werden.
- Aus theoretischer Sicht ist nicht entscheidend, dass die netzwerkseitigen Voraussetzungen und Leistungsgrößen *tatsächlich* am betreffenden IT-System geprüft und gemessen werden, sondern dass dies *prinzipiell* möglich ist. Somit widerspricht es auch nicht dem operationalen Prinzip, wenn beispielsweise das System noch nicht existiert, der fragliche Zeitraum in der Zukunft liegt oder kein geeignetes Messinstrument zur Verfügung steht.
- Stochastische Voraussetzungen, zum Beispiel, widersetzen sich prinzipiell einer experimentellen Prüfung am System und sind daher nicht-operational. Eine Voraussetzung, die man dagegen im Prinzip durchaus experimentell prüfen kann, lautet, dass jede Netzwerkstation im besagten Zeitraum von genauso vielen Jobs betreten und verlassen wird. Sie ist daher operational.
- Weil die Erfüllungsgrade von operationalen Voraussetzungen quantifizierbar sind, besteht die Möglichkeit von quantitativen Fehler- bzw. Abweichungsanalysen der unter den jeweiligen Voraussetzungen geltenden Gesetzmäßigkeiten.
- Sämtliche Betrachtungen eines operationalen Warteschlangenmodells beziehen sich auf einen bestimmten Zeitraum. Daher ist es unerheblich, wie sich das Netzwerk außerhalb dieses Zeitraumes verhält.

Wie schon im Vorwort angedeutet wurde, sind dies ganz wesentliche Punkte, durch die sich operationale von stochastischen Warteschlangenmodellen unterscheiden.

Operationales Warteschlangennetzwerk. Vor diesem Hintergrund ist nun ein operationales Warteschlangennetzwerk – als abstrakte Darstellung eines IT-Systems – in seiner allgemeinsten Form wie folgt definiert (siehe Abbildung 1.1):

- Der Grundbaustein des Netzwerkes ist die *Bedienstation* oder *Multiprozessorstation*. Sie umfasst einen *Wartebereich (Warteraum)*, in welchem Jobs gleichzeitig

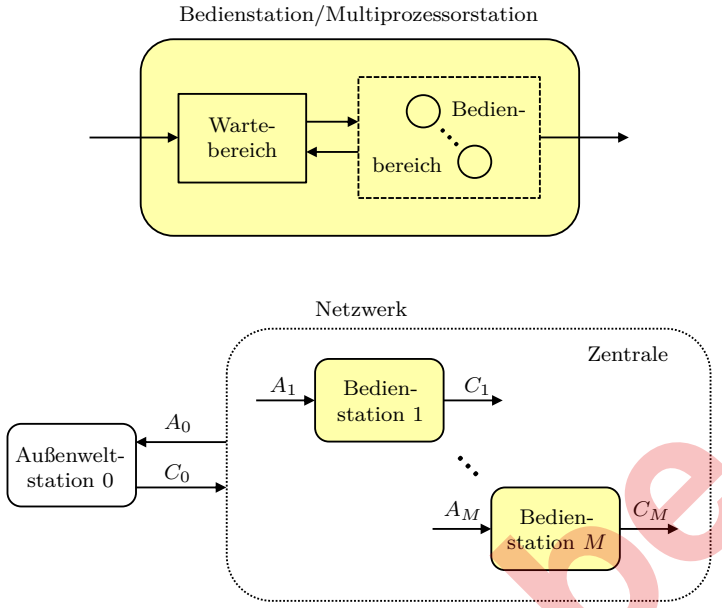


Abbildung 1.1. Allgemeines operationales Warteschlangennetzwerk, bestehend aus einer Schaltung von M Bedienstationen, die zusammen die Zentrale bilden, und einer Außenweltstation. Das Netzwerk ist geschlossen, wenn der äußere Jobeingangskanal mit dem äußeren Jobausgangskanal in der Außenweltstation direkt oder indirekt verbunden ist; ansonsten ist es offen.

auf Verarbeitung warten, und einen *Bedienbereich (Bedienraum)* mit einem oder mehreren Prozessoren, wobei jeder Prozessor nur einen Job gleichzeitig verarbeiten kann. Jobs betreten die Station über den Wartebereich, wechseln zwischen Warte- und Bedienbereich hin und her und verlassen anschließend die Station über den Bedienbereich. Jeder Job wird zum frühstmöglichen Zeitpunkt prozessiert *(Prinzip der Arbeitserhaltung)*. Somit befindet sich die Station genau dann in Betrieb, wenn sie nicht leer ist.

- In den meisten Anwendungsfällen sind die konkreten Warte- und Verarbeitungsmechanismen innerhalb einer Station irrelevant. Hier genügt die Vorstellung, dass einige Jobs irgendwo in der Station warten und andere Jobs irgendwo anders in der Station verarbeitet werden.
- Das Netzwerk selbst besteht aus einer Schaltung von M Bedienstationen $i \in [1:M]$, die zusammen die *Zentrale* bilden, und einer *Außenweltstation* $i = 0$, durch welche die Zentrale mit Jobs versorgt wird. Jobs gelangen von der Außenweltstation in die Zentrale, besuchen und durchlaufen ihre Bedienstationen auf möglicherweise unterschiedlichen Pfaden und verlassen die Zentrale anschließend wieder in Richtung Außenweltstation. Die Jobwanderung von einer Station zur nächsten verläuft verzögerungsfrei.
- In Abhängigkeit vom gewählten Beobachtungszeitraum und von der Jobpropagation durch das Netzwerk ergeben sich für jede Station $i \in [0:M]$ eine eigene Eintrittszahl A_i und eine eigene Austrittszahl C_i. Diese Zahlen charakterisieren die Stationsschaltung bzw. Topologie des Netzwerkes im Beobachtungszeitraum.

- Das Netzwerk ist *offen* oder *transaktional*, wenn die Außenwelt aus einer Jobquelle und einer Jobsenke besteht und somit der äußere Jobeingangs- und Jobausgangskanal dort nicht verbunden sind.
- Das Netzwerk heißt *direkt geschlossen* oder *Batchnetzwerk*, wenn der äußere Jobeingangs- und Jobausgangskanal in der Außenweltstation direkt verbunden sind (wenn also die Außenweltstation kurzgeschlossen ist).
- Das Netzwerk heißt *indirekt geschlossen* oder *interaktiv*, wenn die Außenweltstation eine besondere Art von Bedienstation ist, nämlich eine *Verzögerungs-* bzw. *Terminalstation* (die Erklärung erfolgt in Abschnitt 1.6). Im Grenzfall einer unendlich schnellen Verzögerungsstation geht ein interaktives Netzwerk in ein Batchnetzwerk über.
- Im offenen Fall ist die Zahl C_0 der im Beobachtungszeitraum von der Außenwelt in die Zentrale eintretenden Jobs vorgeben. Im geschlossenen Fall zirkuliert eine fest vorgegebene Zahl N von Jobs durch das Netzwerk.[2]
- Bei Anwesenheit von mehreren Jobklassen ist auch der Fall vorstellbar, dass das Netzwerk bezüglich einiger Jobklassen offen und bezüglich anderer Jobklassen geschlossen ist.
- Zwischen Netzwerk und IT-System existieren folgende Entsprechungen:
  - Netzwerk $\leftrightarrow$ IT-System
  - Zentrale $\leftrightarrow$ IT-Zentralsystem
  - Außenweltstation $\leftrightarrow$ Lasttreiber des IT-Zentralsystems
  - Bedienstation $\leftrightarrow$ Teil des IT-Zentralsystems
  - Warteraum $\leftrightarrow$ Software-Warteschlange
  - Prozessor $\leftrightarrow$ Hardware-Ressource
  - Job $\leftrightarrow$ Anfrage, Auftrag, Transaktion, Zugriff etc.

Aufgrund dieser Beschreibung ist klar, dass eine normale Bedienstation zunächst jeden beliebigen Teilbereich des IT-Zentralsystems repräsentieren kann. Welchen Bereich sie tatsächlich repräsentiert, hängt von ihrem Abstraktionsgrad und ihrer Stellung innerhalb des Netzwerkes ab. Betrachten wir als Beispiel die Abbildung 1.2, in der ein einfacher Rechner durch drei unterschiedlich detaillierte offene Netzwerke dargestellt wird. Auf der niedrigsten Detailebene $M = 1$ besteht die Zentrale aus einer einzigen, indexlosen Station, die den gesamten Rechner repräsentiert. Auf der nächsthöheren Detailebene $M = 2$ haben wir zwei Stationen 1 und 2, von denen die eine für die CPU und die andere für den I/O-Bereich des Rechners steht. Auf der höchsten Detailebene $M = 3$ umfasst die Zentrale die drei Stationen 1, 21 und 22, wobei die letzten beiden jeweils eine Disk des I/O-Komplexes verkörpern.

Eine bequemere Art der Betrachtung von unterschiedlich detaillierten Netzwerken desselben IT-Systems besteht in deren Überlagerung. Dies führt in unserem Beispiel zu Abbildung 1.3. Besonders wichtig hierbei ist jedoch, dass die verschiedenen Netzwerke (Detailebenen) gedanklich immer getrennt bleiben, weil mit jedem Netzwerk

[2] Die korrekte Vorstellung im geschlossenen Fall ist, dass jeder Job beim Verlassen der Außenweltstation durch einen neuen Job (mit neuen Verarbeitungsanforderungen) ersetzt wird.

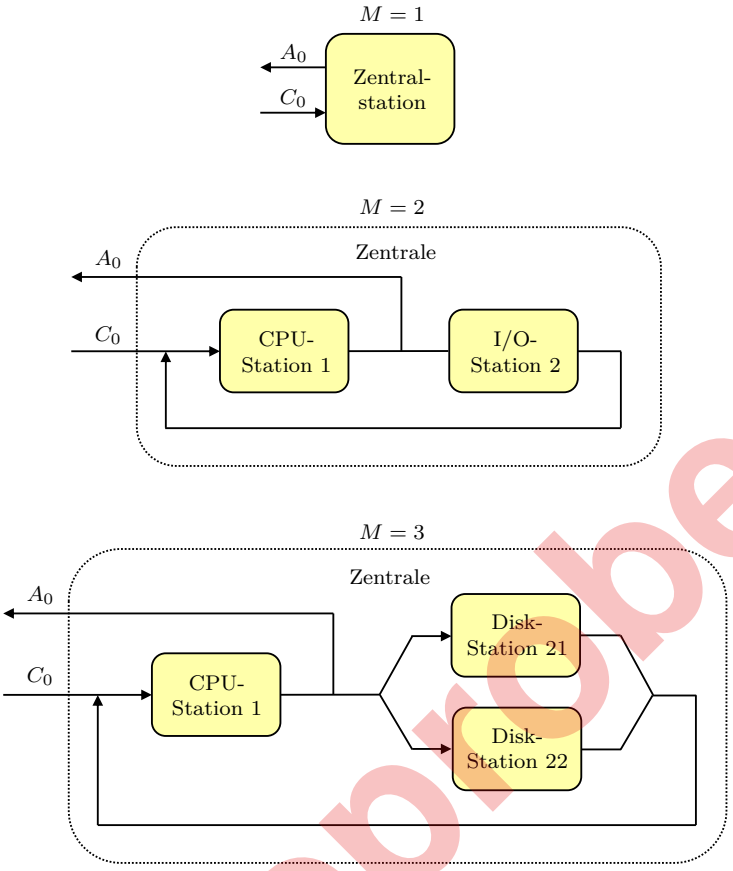


Abbildung 1.2. Drei unterschiedlich detaillierte operationale Netzwerke desselben Rechners (ohne Außenweltstation).

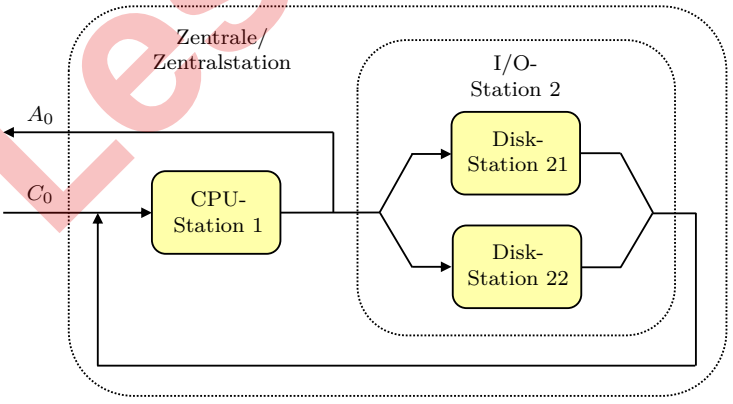


Abbildung 1.3. Überlagerung von drei unterschiedlich detaillierten operationalen Netzwerken desselben Rechners (vgl. Abbildung 1.2).

eine eigene Interpretation der operationalen Voraussetzungen und Leistungsgrößen verbunden ist.

Allgemeine operationale Voraussetzungen. Neben weiteren Voraussetzungen, die wir später einführen werden, gibt es vier Grundvoraussetzungen, die mit jedem Netzwerk automatisch einhergehen. Zusammengefasst lauten sie wie folgt:

Voraussetzung 1.2: Überlappungs- und Blockierungsfreiheit, operationale Verbundenheit und Jobflussgleichgewicht

Gegeben sei das Netzwerk in Abbildung 1.1.

(1) Es findet keine zeitlich überlappende Nutzung von mehreren Stationen durch ein und denselben Job statt *(Überlappungsfreiheit)*.

(2) Die Jobwanderung durch eine Station wird durch keine andere Station blockiert *(Blockierungsfreiheit)*.

(3) Innerhalb des Beobachtungszeitraumes wird jede Station von wenigstens einem, von außen kommenden Job direkt oder indirekt besucht *(operationale Verbundenheit)*.

(4) Innerhalb des Beobachtungszeitraumes sind die Eintritts- und Austrittszahlen einer jeden Station identisch: $A_i = C_i \; \forall\, i \in [0:M]$ *(Jobflussgleichgewicht)*.

Die erste Voraussetzung bedeutet, dass jeder Job einzelne Stationen strikt nacheinander durchläuft und somit in eindeutiger Weise durch das Netzwerk *wandert*. Wie sich in der Praxis zeigt, ist dies nicht sehr einschränkend, insbesondere auf der Ebene der Inter-Serverkommunikation. Aber auch innerhalb eines Servers beträgt die Überlappung von CPU und I/O typischerweise höchstens einige Prozent.

Die zweite Voraussetzung ist dagegen etwas problematischer und sollte im Zweifel geprüft werden. In einem Server können beispielsweise Blockaden entstehen, wenn ein I/O-Buffer vollläuft und die CPU deshalb ausgebremst wird.

Die dritte Voraussetzung bringt zum Ausdruck, dass das Netzwerk hinsichtlich seines Jobflusses nicht in unabhängige Subnetze unterteilt werden kann. Dies stellt in der Praxis kein Problem dar, sofern der Beobachtungszeitraum hinreichend groß gewählt wird und das IT-System hardware- und softwaremäßig verbunden ist.

Die vierte Voraussetzung ist das Herzstück der operationalen Warteschlangentheorie. Sie ermöglicht es, die ein- und austretenden Jobströme aller Stationen mathematisch in Beziehung zu setzen. Bei großen Beobachtungszeiträumen ist das Jobflussgleichgewicht sowohl auf Netzwerk- als auch auf IT-Systemebene meistens näherungsweise erfüllt, weil die relative Abweichung $|A_i - C_i|/C_i$ mit der Zeit tendenziell kleiner wird, sofern der Verarbeitungsrückstand (aufgrund von permanenter Überlastung) nicht ständig zunimmt.

Schlussbemerkung. Im weiteren Verlauf werden wir der Einfachheit halber meistens auf die konsequente sprachliche Trennung von IT-System- und Modellbegriffen verzichten und beispielsweise sowohl ein IT-System als auch das mit ihm assoziierte modellhafte Netzwerk mit „System“ bezeichnen. Umso wichtiger ist es, im Auge zu behalten, dass wir uns streng genommen immer auf der Modell- bzw. Netzwerkebene bewegen. Desweiteren ist im Regelfall davon auszugehen, dass (i) der Warteraum einer jeden Bedienstation unbeschränkt ist und (ii) genau eine Job-

klasse existiert. Schließlich treffen wir auch noch die Vereinbarung, dass sämtliche Größen, die einen Stationsindex tragen, in derselben Weise auch ohne Stationsindex definiert sind und sich dann auf die gesamte Zentrale als eigenständige Bedienstation (Detailebene $M = 1$) beziehen.

1.2 Leistungsgrößen

Nehmen wir das allgemeine System in Abbildung 1.1 inklusive der Voraussetzung 1.2. An all seinen Stationen $i \in [0 : M]$ können im Prinzip folgende Basisgrößen gemessen werden:[3]

$T \equiv$ Beobachtungs- oder Messzeitraum
$A_i \equiv$ Zahl der Jobeintritte in die i-te Station
$C_i \equiv$ Zahl der Jobaustritte aus der i-ten Station
$F_i \equiv$ akkumulierte Jobverweilzeit in der i-ten Station
$B_i \equiv$ Betriebszeit (Nicht-Leerzeit) der i-ten Station.

Hierüber lassen sich einige grundlegende Leistungsgrößen definieren:

Definition 1.3: Fundamentale Leistungsgrößen
Gegeben sei das System in Abbildung 1.1. Für jede Station $i \in [0 : M]$ soll gelten:

$$V_i = \frac{C_i}{C} \equiv \text{mittleres Besuchsverhältnis } (\Rightarrow V_0 = V = 1)$$

$$\lambda_i = \frac{A_i}{T} \equiv \text{mittlere Eintrittsrate}$$

$$X_i = \frac{C_i}{T} \equiv \text{mittlere Austrittsrate} \equiv \text{mittlerer Durchsatz } (\Rightarrow X_i = \lambda_i)\,.$$

Für jede Bedienstation $i \in [0/1 : M]$ soll gelten:[4]

$$Q_i = \frac{F_i}{T} \equiv \text{mittlere Jobzahl}$$

$$R_i = \frac{F_i}{C_i} \equiv \text{mittlere Antwortzeit (pro Stationsbesuch)}$$

$$\mu_i = \frac{C_i}{B_i} \equiv \text{mittlere Bedienrate}$$

$$S_i = \frac{B_i}{C_i} = \frac{1}{\mu_i} \equiv \text{mittlere Bedienzeit (pro Stationsbesuch)}$$

$$D_i = \frac{B_i}{C} = V_i S_i \equiv \text{mittlerer Bedienaufwand (pro Zentralbesuch)}$$

[3] Im nicht-interaktiven Fall gilt $F_0 = B_0 = 0$.
[4] $[0/1 : M]$ bedeutet $[0 : M]$ im interaktiven Fall und $[1 : M]$ im nicht-interaktiven Fall.

$$U_i = \frac{B_i}{T} \equiv \text{mittlere Auslastung}\,.$$

Zum korrekten Verständnis dieser Festlegungen beachte man:

Mittelwerte. Bei sämtlichen Leistungsgrößen handelt es sich um Mittelwerte, die ausschließlich den Beobachtungszeitraum T betreffen.

Mittleres Besuchsverhältnis V_i. Dieses Verhältnis gibt an, wieviele Besuche der Station i mit einem Besuch der Zentrale im Mittel verbunden sind. Die Gesamtheit aller Besuchsverhältnisse definiert die Topologie des Systems.

Mittlerer Durchsatz X_i. Von einem Durchsatz im gewohnten Sinne kann in der Tat nur die Rede sein, wenn die betreffende Station im Messzeitraum von gleich vielen Jobs betreten und verlassen wird (Jobflussgleichgewicht). Ist die Station am Anfang und Ende der Messung leer, dann umfasst der Durchsatz ausschließlich komplette Stationsdurchläufe.

Mittlere Jobzahl Q_i. Ihre Formel erklärt sich wie folgt: Gemäß Abbildung 1.4 ist die akkumulierte Jobverweilzeit F_i gleich der Fläche unterhalb der Kurve der momentanen Stationsjobzahl bzw. momentanen Stationsbesetzung $n_i(t)$ im Messzeitraum T. Somit entspricht Q_i der mittleren Flächenhöhe F_i/T.

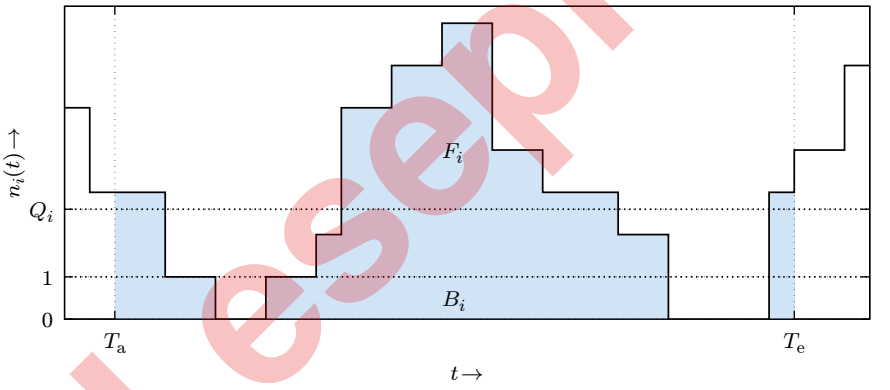


Abbildung 1.4. Momentane Besetzung einer Bedienstation i als Funktion der Zeit. Der Messzeitraum $[T_a : T_e]$ wurde so gewählt, dass das Jobflussgleichgewicht $A_i = C_i \Leftrightarrow n_i(T_a) = n_i(T_e)$ erfüllt ist. Die blaue Fläche unterhalb der Kurve entspricht der akkumulierten Jobverweilzeit F_i, und die blaue Fläche unterhalb der Eins entspricht der Betriebszeit B_i. Die mittlere Jobzahl Q_i ist gleich der mittleren Flächenhöhe F_i/T, mit $T = T_e - T_a$.

Mittlere Antwortzeit R_i. Im allgemeinen entspricht diese Größe nicht der herkömmlichen Variante von mittlerer Antwortzeit, wie man sie durch Mittelung aller Jobverweilzeiten vom Stationseintritt bis zum Stationsaustritt erhält. Der Grund hierfür sind die partiellen Verweilzeiten an den Rändern des Messzeitintervalls, wie in Abbildung 1.4 zu sehen. Immerhin stimmen beide Antwortzeitvarianten näherungsweise überein, wenn $n_i(T_a) \approx n_i(T_e) \ll C_i$, was bei großen Messzeiträumen meistens der Fall ist. Die Übereinstimmung ist exakt, falls die Station am Anfang

und Ende der Messung leer ist: $n_i(T_a) = n_i(T_e) = 0$. Ähnliche Überlegungen gelten auch für die anderen beiden Zeitgrößen S_i und D_i.

Mittlere Bedienrate μ_i. Im Gegensatz zur Austrittsrate berücksichtigt diese Größe ausschließlich die Betriebsphasen der betreffenden Station. Unter der Bedingung $A_i = C_i$ folgt (wegen $B_i \leq T$)

$$\lambda_i = \frac{A_i}{T} = \frac{C_i}{T} \leq \frac{C_i}{B_i} = \mu_i \,.$$

Somit stellt $\lambda_i \leq \mu_i$ ein notwendiges (aber nicht hinreichendes) Kriterium für Jobflussgleichgewicht dar.

Mittlere Bedienzeit S_i. Im allgemeinen ist die Bedienzeit kleiner oder gleich der Prozessorzeit, die während der Jobverarbeitung anfällt. Bedienzeit und Prozessorzeit stimmen genau dann überein, wenn die Station zu keiner Zeit mehrere Jobs gleichzeitig verarbeitet.

Mittlerer Bedienaufwand D_i. Er ist gleich der Bedienzeit der Station i, die bei einem Besuch der Zentrale – unter Berücksichtigung der damit einhergehenden V_i-fachen Stationsbesuche – im Mittel aufgewendet wird.

Mittlere Auslastung U_i. Generell ist die Auslastung einer Station kleiner oder gleich der Summe der Auslastungen ihrer Prozessoren (siehe Abbildung 1.5). Gleichheit besteht genau dann, wenn in der Station immer höchstens ein Job gleichzeitig verarbeitet wird.[5]

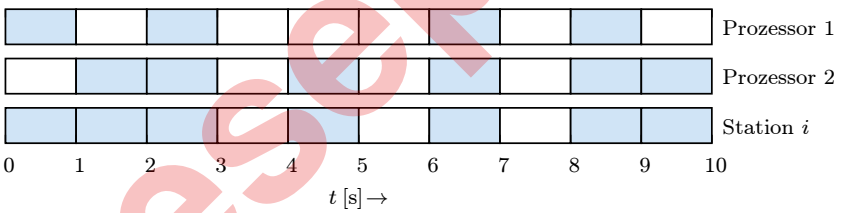


Abbildung 1.5. Auslastungsbilanz einer Zweiprozessorstation i. Im Messzeitraum $[0:10\text{s}]$ sind die Prozessoren zu 40% bzw. 60% ausgelastet. Die Stationsauslastung $U_i = 70\%$ resultiert aus der Überlagerung der einzelnen Prozessorzeiten und nicht aus deren Summe.

Schlussbemerkung. Bei den in den folgenden Abschnitten aus Definition 1.3 abgeleiteten Gesetzmäßigkeiten ist zu beachten, dass manche von ihnen (nämlich Littles Gesetz, das Durchsatzgesetz, das allgemeine Antwortzeitgesetz und das Auslastungsgesetz) auch ohne die Voraussetzung des Jobflussgleichgewichtes gelten. Trotzdem ist ihre korrekte Anwendung in vielen Fällen an das Jobflussgleichgewicht gebunden.

[5] Inhaltlich korrektere Bezeichnungen für S_i, D_i und U_i wären *mittlere Betriebszeit*, *mittlerer Betriebsaufwand* und *mittlerer Betriebszeitanteil*. Sie haben sich jedoch nicht durchgesetzt. In Abschnitt 1.7 werden wir die mit Hut versehenen Größen $\hat{S}_i \equiv$ *mittlere Prozessorzeit*, $\hat{D}_i \equiv$ *mittlerer Prozessoraufwand* und $\hat{U}_i \equiv$ *mittlere effektive Auslastung* einführen.

1.3 Littles Gesetz

Aus den Definitionen von Q_i, R_i und X_i folgt

$$Q_i = \frac{F_i}{T} = \frac{F_i}{C_i}\frac{C_i}{T} = R_i X_i \,.$$

Somit lautet unser erstes und wichtigstes operationales Gesetz:[6]

Satz 1.4: Littles Gesetz
Gegeben sei das System in Abbildung 1.1. Für jede Bedienstation $i \in [0/1 : M]$ besteht zwischen der mittleren Jobzahl Q_i, der mittleren Antwortzeit R_i und dem mittleren Durchsatz X_i der Zusammenhang

$$Q_i = R_i X_i \,.$$

Wie wir im weiteren Verlauf des Buches schnell sehen werden, stellt dieses Gesetz eine wesentliche Grundlage der gesamten operationalen Warteschlangentheorie dar. Ein unmittelbar einsichtiger Vorzug von Littles Gesetz besteht darin, dass es – bei entsprechender Wahl der Detailebene und entsprechender Interpretation der involvierten Größen – auf alle Systembereiche angewendet werden kann. Hierzu zählen beispielsweise die gesamte Zentrale, einzelne Stationen innerhalb der Zentrale, die Warte- und Bedienbereiche innerhalb einer Station sowie einzelne Prozessoren innerhalb eines Bedienbereiches.

Beispielrechner. Um die Anwendung von Littles Gesetz zu demonstrieren, betrachten wir den Rechner in Abbildung 1.6, der auch in nachfolgenden Abschnit-

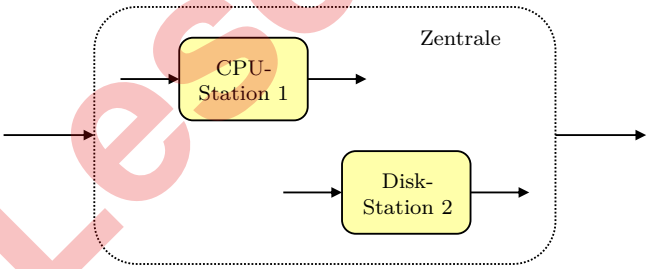


Abbildung 1.6. Beispielrechner mit zwei Bedienstationen.

ten als Beispiel herangezogen wird. Hierbei beziehen sich indizierte Größen auf die Stationen 1 und 2 (Detailebene $M = 2$), während indexlose Größen den Rechner selbst betreffen (Detailebene $M = 1$). Angenommen, es werden im selben Messzeitraum die Werte $Q_1 = 10$, $X_1 = 500/\text{s}$, $R_2 = 80\text{ms}$, $X_2 = 25/\text{s}$ und $X = 50/s$ gemessen, dann liefert Littles Gesetz für genau diesen Zeitraum eine mittlere Antwortzeit der ersten Station von $R_1 = Q_1/X_1 = 0.02\text{s}$, eine mittlere Jobzahl der

[6] Dieses Gesetz ist das operationale Pendant zu einem Gesetz, das im Jahre 1961 von John Little im stochastischen Kontext erstmalig bewiesen wurde.

zweiten Station von $Q_2 = R_2 X_2 = 2$ sowie eine mittlere Zentralantwortzeit von $R = Q/X = (Q_1 + Q_2)/X = 0.24$s.

1.4 Durchsatzgesetz

Noch einmal ausgehend von Definition 1.3 lässt sich der Durchsatz einer Station i mit dem Zentraldurchsatz folgendermaßen in Verbindung bringen:

$$X_i = \frac{C_i}{T} = \frac{C_i}{C}\frac{C}{T} = V_i X \,.$$

Hieraus folgt:

Satz 1.5: Durchsatzgesetz

Gegeben sei das System in Abbildung 1.1. Für jede Station $i \in [0 : M]$ besteht zwischen dem mittleren Durchsatz X_i, dem mittleren Besuchsverhältnis V_i und dem mittleren Zentraldurchsatz X der Zusammenhang

$$X_i = V_i X \,.$$

Eine wichtige Aussage dieses Gesetzes ist, dass aus der Gesamtheit aller Besuchsverhältnisse und dem Durchsatz einer Station die Durchsätze aller anderen Stationen folgen. Bezugnehmend auf den Beispielrechner in Abbildung 1.6 mit den oben genannten Durchsätzen $X_1 = 500/\text{s}$, $X_2 = 25/\text{s}$ und $X = 50/\text{s}$ folgen aus dem Durchsatzgesetz die Besuchsverhältnisse $V_1 = X_1/X = 10$ und $V_2 = X_2/X = 0.5$, mit der Bedeutung, dass im Messzeitraum jeder Zentralbesuch im Mittel mit 10 Besuchen von Station 1 und 0.5 Besuchen von Station 2 einherging.

1.5 Allgemeines Antwortzeitgesetz

Intuitiv ist klar, dass man aus den Antwortzeiten und Besuchsverhältnissen der zentralinternen Bedienstationen die Antwortzeit der Zentrale berechnen können sollte. Die formale Rechtfertigung hierfür lautet

$$R = \frac{Q}{X} = \frac{1}{X}\sum_{i=1}^{M} Q_i = \frac{1}{X}\sum_{i=1}^{M} R_i X_i = \sum_{i=1}^{M} R_i V_i \,,$$

wobei Littles Gesetz zweimal und das Durchsatzgesetz einmal angewendet wurden.

Satz 1.6: Allgemeines Antwortzeitgesetz

Gegeben sei das System in Abbildung 1.1. Die mittlere Zentralantwortzeit R ist gleich der Summe der mittleren Antwortzeiten R_i aller zentralinternen Bedienstationen, gewichtet mit den zugehörigen mittleren Besuchsverhältnissen V_i:

$$R = \sum_{i=1}^{M} R_i V_i \,.$$

Im Fall unseres Beispielrechners mit den Werten $R_1 = 0.02\text{s}$, $V_1 = 10$, $R_2 = 80\text{ms}$ und $V_2 = 0.5$ liefert dieses Gesetz die bereits bekannte mittlere Zentralantwortzeit von $R = R_1 V_1 + R_2 V_2 = 0.24\text{s}$.

1.6 Interaktives Antwortzeitgesetz

Abbildung 1.7 zeigt eine IT-Konstellation, in der gleichzeitig mehrere Nutzer über eigene Terminals mit einem Rechenzentrum interagieren. Ein allgegenwärtiges Bei-

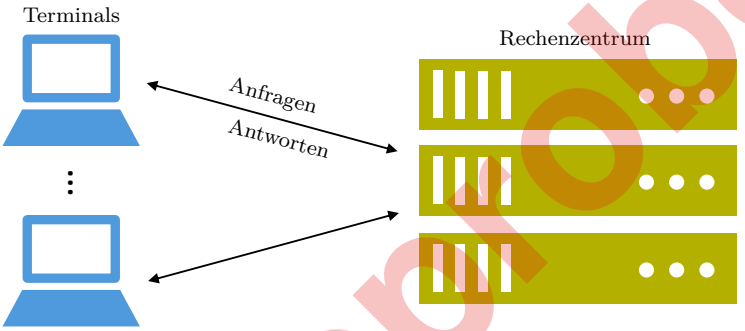


Abbildung 1.7. Interaktives IT-System. Nutzer kommunizieren via Terminals mit einem Rechenzentrum. Dessen Leistungsverhalten hängt unter anderem von der Zahl der Nutzer sowie von den Pausen (Denkzeiten) ab, die jeder Nutzer zwischen dem Erhalt einer Antwort und dem Senden der nächsten Anfrage einlegt.

spiel dieser Art ist das Online-Banking: Nutzer verbinden sich über ihren Browser (oder ihre Banking-App) mit dem Rechenzentrum, loggen sich mit ihren Zugangsdaten ein, fragen ihre Kontostände ab, führen Überweisungen durch, lesen aktuelle Kundeninformationen und loggen sich anschließend wieder aus. Typischerweise geht mit jeder Terminalaktion (Link anklicken, Button drücken etc.) eine Anfrage an das Rechenzentrum einher, die umgehend beantwortet wird. Zwischen dem Erhalt einer Antwort und dem Senden der nächsten Anfrage befindet sich jeder Nutzer in einem *Denkzustand*, in welchem er das Resultat seiner letzten Aktion in Augenschein nimmt und seine nächste Aktion vorbereitet.

Eine naheliegende Abstraktion dieses Sachverhaltes ist in Abbildung 1.8 dargestellt. Dort sehen wir ein geschlossenes System mit zwei Stationen, wobei die *Terminal-* oder *Verzögerungsstation* (ehemals Außenweltstation) sämtliche Nutzerterminals und die Zentralstation das Rechenzentrum repräsentiert. Die hiermit verbundene Vorstellung ist, dass insgesamt N Jobs (ein Job pro Terminal) im System zirkulieren und bei jedem Rundgang im Mittel die *Denkzeit* Z in der Terminalstation und die Antwortzeit R in der Zentralstation verbringen. Hierfür liefert nun Litt-

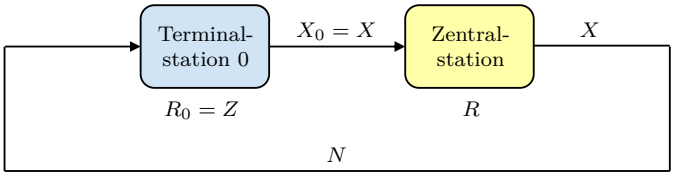


Abbildung 1.8. Interaktives (geschlossenes) System mit einer Zentralstation und einer Terminalstation als Außenwelt. N ist die Zahl der im System zirkulierenden Jobs, $R_0 = Z$ die mittlere Antwortzeit (Denkzeit) und X_0 der mittlere Durchsatz der Terminalstation, R die mittlere Antwortzeit und X der mittlere Durchsatz der Zentralstation. Unter der Voraussetzung des Jobflussgleichgewichtes gilt $X_0 = X$. Der Spezialfall $Z = 0$ entspricht einem Batchsystem.

les Gesetz in Kombination mit dem Jobflussgleichgewicht und dem allgemeinen Antwortzeitgesetz

$$N = X(Z + R) \Leftrightarrow R = \frac{N}{X} - Z\,.$$

Grundsätzlich ist diese Beziehung natürlich nicht daran gebunden, dass eine der beiden Stationen lastunabhängige Antwortzeiten (Denkzeiten) besitzt. Sie wird jedoch in diesem Sinne am häufigsten verwendet.

Satz 1.7: Interaktives Antwortzeitgesetz
Gegeben sei das System in Abbildung 1.8. Dann gilt der Zusammenhang

$$R = \frac{N}{X} - Z\,,$$

wobei N die konstante Jobzahl im System, Z die mittlere Denkzeit der Terminalstation, X den mittleren Durchsatz und R die mittlere Antwortzeit der Zentralstation bezeichnen.

Dieses Gesetz findet vor allem bei *geschlossenen Performancetests* Anwendung, deren experimenteller Aufbau in struktureller Hinsicht genau der Abbildung 1.7 entspricht und daher durch Abbildung 1.8 abstrahiert werden kann. Die Terminalnutzer werden dabei durch *virtuelle User* simuliert, die ihre Anfragen skriptgesteuert an das Rechenzentrum senden und die zugehörigen Antwortzeiten messen. Nehmen wir jetzt an, die im Zusammenhang mit unserem Beispielrechner weiter oben genannten Leistungswerte $X = 50/\text{s}$, $R = 0.24\text{s}$ und $Q = 12$ resultierten aus einem geschlossenen Performancetest gegen diesen Rechner, und zwar mit $N = 112$ virtuellen Usern. Dann folgt aus dem interaktiven Antwortzeitgesetz, dass alle User während des Tests eine mittlere Denkzeit von $Z = N/X - R = 2\text{s}$ zwischen dem Empfang einer Antwort und dem Senden der nächsten Anfrage eingelegt haben. Ferner befanden sich im Mittel $Q_0 = N - Q = 100$ User gleichzeitig im Denkzustand.

1.7 Auslastungsgesetz

Aufgrund von Definition 1.3 erhalten wir für die mittlere Auslastung U_i, den mittleren Durchsatz X_i und die mittlere Bedienzeit S_i einer Bedienstation i den Zusammenhang

$$U_i = \frac{B_i}{T} = \frac{B_i}{C_i}\frac{C_i}{T} = S_i X_i \,. \tag{1.1}$$

Wie in den Bemerkungen zu Definition 1.3 bereits festgestellt wurde, stimmt S_i im allgemeinen nicht mit der mittleren Prozessorzeit pro Stationsbesuch überein. Dies geht damit einher, dass U_i im allgemeinen nicht die Prozessorzeiten aller Stationsprozessoren in Summe umfasst. Um nun einen stationsseitigen Auslastungszusammenhang zu entwickeln, der die mittlere Prozessorzeit (≡ mittlere Jobverweilzeit im Bedienbereich) enthält, nehmen wir an, dass die Station i genau m_i Prozessoren besitzt, wobei im Beobachtungszeitraum T für jeden Prozessor $k \in [1 : m_i]$ die Betriebszeit B_{ik} und somit die mittlere Auslastung $U_{ik} = B_{ik}/T$ gemessen werden. Damit folgt aufgrund einer ähnlichen Rechnung wie in (1.1)

$$\hat{U}_i = \sum_{k=1}^{m_i} U_{ik} = \sum_{k=1}^{m_i} \frac{B_{ik}}{T} = \sum_{k=1}^{m_i} \frac{B_{ik}}{C_i}\frac{C_i}{T} = \hat{S}_i X_i \;,\; \hat{S}_i = \sum_{k=1}^{m_i} \frac{B_{ik}}{C_i} \,,$$

wobei $\hat{U}_i$ die stationsseitige *mittlere effektive Auslastung* (≡ Summe der mittleren Prozessorauslastungen) und $\hat{S}_i$ die stationsseitige *mittlere Prozessorzeit (pro Stationsbesuch)* bezeichnen. Weil die Betriebszeiten B_{ik} der Prozessoren mit deren akkumulierten Jobverweilzeiten F_{ik} übereinstimmen, erhalten wir zusätzlich für die mittleren Jobzahlen $Q_i^{(\mathrm{s})}$ und $Q_i^{(\mathrm{w})}$ der stationsseitigen Bedien- und Wartebereiche

$$Q_i^{(\mathrm{s})} = \sum_{k=1}^{m_i} \frac{F_{ik}}{T} = \sum_{k=1}^{m_i} \frac{B_{ik}}{T} = \hat{U}_i \;,\; Q_i^{(\mathrm{w})} = Q_i - Q_i^{(\mathrm{s})} \,.$$

Insgesamt können wir somit festhalten:

Satz 1.8: Auslastungsgesetz

Gegeben sei das System in Abbildung 1.1. Für jede Bedienstation $i \in [0/1 : M]$ besteht zwischen der mittleren Auslastung U_i, der mittleren Bedienzeit S_i, dem mittleren Bedienaufwand D_i, dem mittleren Besuchsverhältnis V_i, dem mittleren Durchsatz X_i und dem mittleren Zentraldurchsatz X der Zusammenhang

$$U_i = S_i X_i = D_i X \;,\; D_i = V_i S_i \,. \tag{1.2}$$

Desweiteren gilt

$$\hat{U}_i = Q_i^{(\mathrm{s})} = \sum_{k=1}^{m_i} U_{ik} = \hat{S}_i X_i = \hat{D}_i X \;,\; \hat{D}_i = V_i \hat{S}_i = \sum_{k=1}^{m_i} D_{ik} \;,\; D_{ik} = \frac{B_{ik}}{C} \,, \tag{1.3}$$

mit

$m_i \equiv$ Prozessorzahl
$Q_i^{(s)} \equiv$ mittlere Jobzahl des Bedienbereiches
$\hat{U}_i \equiv$ mittlere effektive Auslastung (mit Werten im Bereich $[0 : m_i]$)
$\hat{S}_i \equiv$ mittlere Prozessorzeit (pro Stationsbesuch)
$\hat{D}_i \equiv$ mittlerer Prozessoraufwand (pro Zentralbesuch)
$U_{ik} \equiv$ mittlere Auslastung des k-ten Prozessors
$D_{ik} \equiv$ mittlerer Bedienaufwand des k-ten Prozessors (pro Zentralbesuch)
$B_{ik} \equiv$ mittlere Betriebszeit des k-ten Prozessors.

Bei einer Einprozessorstation ($m_i = 1$) fallen (1.2) und (1.3) zusammen.

Wie man sich leicht klarmacht, folgen hieraus die Ungleichungen

$$0 \le \hat{U}_i \le m_i U_i \le m_i \min\left\{\hat{U}_i, 1\right\}.$$

Kommen wir ein letztes mal auf unseren Beispielrechner in Abbildung 1.6 zurück und nehmen an, dass die Station 1 eine Zweiprozessorstation ist, an der im Messzeitraum die Auslastungswerte $U_1 = 90\%$, $U_{11} = 40\%$ und $U_{12} = 80\%$ gemessen wurden. Unter Berücksichtigung der bekannten Werte $Q_1 = 10$, $X_1 = 500/\text{s}$ und $R_1 = 0.02\text{s}$ lässt sich dann für diese Station folgendes feststellen:

- Mittlere Bedienzeit: $S_1 = U_1/X_1 = 0.0018\text{s}$.
- Mittlere Bedienrate: $\mu_1 = 1/S_1 \approx 556/\text{s}$.
- Mittlere effektive Auslastung: $\hat{U}_1 = U_{11} + U_{12} = 120\%$.
- Mittlere Prozessorzeit: $\hat{S}_1 = \hat{U}_1/X_1 = 0.0024\text{s}$.
- Mittlere Prozessorrate: $\hat{\mu}_1 = 1/\hat{S}_1 \approx 417/\text{s}$.
- Mittlere Wartezeit: $W_1 = R_1 - \hat{S}_1 = 0.0176\text{s}$.
- Mittlere Zahl der gleichzeitig bedienten Jobs: $Q_1^{(s)} = \hat{S}_1 X_1 = \hat{U}_1 = 1.2$.
- Mittlere Zahl der gleichzeitig wartenden Jobs: $Q_1^{(w)} = W_1 X_1 = Q_1 - Q_1^{(s)} = 8.8$.

Anhand des Verhältnisses $W_1/R_1 = Q_1^{(w)}/Q_1 = 0.88$ ist zu erkennen, dass jeder Job im Mittel 88% seiner stationsseitigen Verweilzeit im Warteraum und 12% in einem Prozessor verbracht hat.

Multiprozessormodul. Wir beenden diesen Abschnitt, indem wir eine sehr einfach strukturierte Multiprozessorstation vorstellen:

Definition 1.9: Multiprozessormodul, m-Prozessormodul
Ein Multiprozessormodul (genauer: m-Prozessormodul) ist eine Multiprozessorstation mit einem Warteraum und m parallel geschalteten Prozessoren (siehe Abbildung 1.9). Das Modul heißt *symmetrisch*, wenn alle seine Prozessoren operational identisch sind.

Die operationale Besonderheit eines Multiprozessormoduls besteht darin, dass sich die vier Zeitgrößen mittlere Bedienzeit S, mittlere Prozessorzeit $\hat{S}$, mittlere Wartezeit W und mittlere Antwortzeit R allein aus seiner Besetzungskurve und seiner Jobaustrittszahl ergeben (bei beliebigen Multiprozessorstationen ist dies nur für S

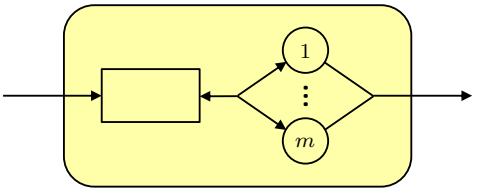


Abbildung 1.9. m-Prozessormodul, bestehend aus einem Warteraum und m parallel geschalteten Prozessoren. Einprozessormodul und Einprozessorstation bedeuten dasselbe.

und R der Fall). Abbildung 1.10 zeigt einen möglichen Verlauf der momentanen Besetzung $n(t)$ eines m-Prozessormoduls im Messzeitraum T, wobei die zugehörige Fläche durch die horizontalen Geraden bei $n = 1$ und $n = m$ in die Segmente F_1, F_2 und F_3 unterteilt ist. Die Fläche F_1 entspricht der Betriebszeit des Moduls. Weil

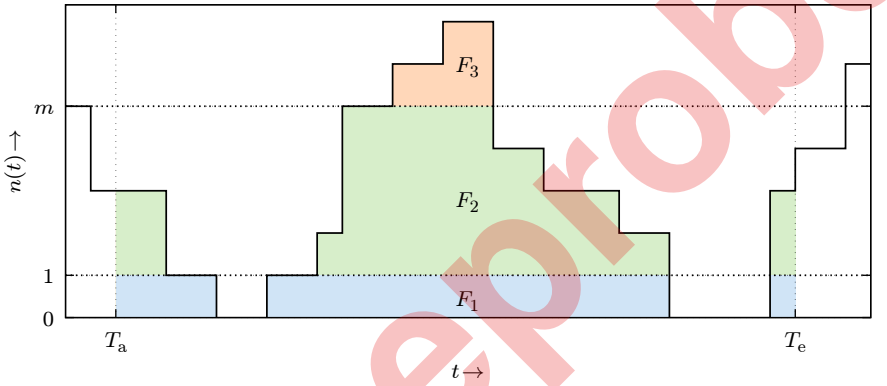


Abbildung 1.10. Momentane Besetzung eines m-Prozessormoduls als Funktion der Zeit. Die innerhalb des Messzeitraumes $[T_a : T_e]$ akkumulierte Jobverweilzeit ist gleich der bunten Fläche unterhalb der Kurve. Diese Fläche wird durch die Geraden bei $n = 1$ und $n = m$ in die drei Segmente F_1, F_2 und F_3 unterteilt.

im Modul erst bei einer Besetzung von $n > m$ gewartet werden muss (Prinzip der Arbeitserhaltung), beschreibt die Fläche $F_1 + F_2$ die akkumulierte Prozessorzeit und F_3 die akkumulierte Wartezeit. Zusammen mit der Jobaustrittszahl C folgt deshalb

$$S = \frac{F_1}{C}\ ,\ \hat{S} = \frac{F_1 + F_2}{C}\ ,\ W = \frac{F_3}{C}\ ,\ R = \frac{F_1 + F_2 + F_3}{C}\ .$$

Multiprozessormodule eignen sich beispielsweise zur Modellierung von IT-Funktionseinheiten wie Mehrkern-CPUs oder Disk-Arrays, aber auch von Verzögerungsphänomenen wie Leitungslatenzen oder Nutzerterminals. So kann etwa die Terminalstation in Abbildung 1.8 als ein symmetrisches Multiprozessormodul mit unendlich vielen (oder mit wenigstens N) Prozessoren und ständig leerem Warteraum aufgefasst werden, wobei jeder Job im Mittel die Denkzeit Z in einem Prozessor verbringt.

1.8 Verkehrsgleichungen

In diesem Abschnitt wollen wir uns in größtmöglicher Allgemeinheit mit der Frage beschäftigen, was die Voraussetzung 1.2 für die stationsseitigen Jobströme des Systems in Abbildung 1.1 bedeutet. Nehmen wir hierzu an, dass im Beobachtungszeitraum T am System die *gerichteten Austrittszahlen*

$$C_{ij} \equiv \left\{ \begin{array}{l} \text{Zahl der Jobaustritte aus der } i\text{-ten} \\ \text{in Richtung der } j\text{-ten Station} \end{array} \right\} \forall\, i,j \in [0:M]\ ,\ C_{00} = 0$$

gemessen werden (siehe Abbildung 1.11). Dann kann das Prinzip des Jobflussgleich-

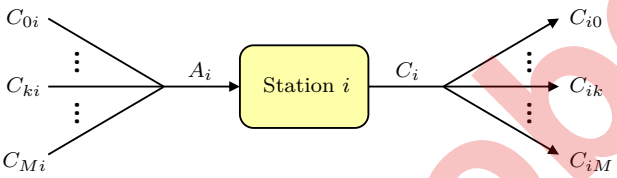


Abbildung 1.11. Gerichtete Austrittszahlen hin zur und weg von der i-ten Station.

gewichtes in der Weise

$$X_i = \frac{C_i}{T} = \frac{A_i}{T} = \sum_{k=0}^{M} \frac{C_{ki}}{T}$$

ausgedrückt werden. Durch Einführung der *mittleren Routingfrequenzen*

$$q_{ij} = \frac{C_{ij}}{C_i}\ \forall\, i,j \in [0:M]\ ,\ q_{00} = 0$$

folgt weiterhin

$$X_i = \sum_{k=0}^{M} X_k q_{ki}\ ,\ \sum_{k=0}^{M} q_{ik} = 1\,, \tag{1.4}$$

wobei die operationale Verbundenheit des Systems die Existenz aller q_{ij} garantiert: $C_i > 0\ \forall\, i \in [0:M]$. Der linke Teil von (1.4) ist ein lineares Gleichungssystem mit $M+1$ Gleichungen für die ebenso vielen Durchsätze X_i. Davon sind allerdings nicht alle Gleichungen linear unabhängig, weil zum Beispiel die Summe der ersten bis M-ten Gleichung gerade die nullte Gleichung ergibt:

$$\sum_{i=1}^{M} X_i = \sum_{k=0}^{M} X_k \sum_{i=1}^{M} q_{ki} = \sum_{k=0}^{M} X_k (1 - q_{k0}) = X_0 + \sum_{k=1}^{M} X_k - \sum_{k=0}^{M} X_k q_{k0}$$

$$\Rightarrow X_0 = \sum_{k=0}^{M} X_k q_{k0}\,.$$

Nun sorgt jedoch die operationale Verbundenheit auch dafür, dass genau M der $M+1$ Gleichungen linear unabhängig sind. Dies wiederum bedeutet, dass bei einem offenen System das Gleichungssystem in (1.4) eine eindeutige Lösung besitzt, weil der äußere Durchsatz X_0 vorgegeben ist. Im Fall eines geschlossenen Systems existiert dagegen nur eine bis auf X_0 bestimmte Lösung. Der Zusammenhang zwischen den Routingfrequenzen q_{ij} und den Besuchsverhältnissen V_i tritt zutage, indem wir jede Gleichung im linken Teil von (1.4) durch den Außen- bzw. Zentraldurchsatz X_0 dividieren und anschließend das Durchsatzgesetz ausnutzen:

$$\frac{X_i}{X_0} = \sum_{k=0}^{M} \frac{X_k}{X_0} q_{ki} \Leftrightarrow V_i = \sum_{k=0}^{M} V_k q_{ki} \,.$$

Wegen $V_0 = 1$ ist dieses Gleichungssystem sowohl für offene als auch für geschlossene Systeme eindeutig lösbar.

Satz 1.10: Verkehrsgleichungen

Gegeben sei das System in Abbildung 1.1. Dann gelten die Verkehrsgleichungen $(i, j \in [0 : M])$

$$X_i = \sum_{k=0}^{M} X_k q_{ki} \;,\; q_{ij} = \frac{C_{ij}}{C_i} \;,\; \sum_{k=0}^{M} q_{ik} = 1$$

und

$$V_i = \sum_{k=0}^{M} V_k q_{ki} \;,\; V_i = \frac{C_i}{C_0} = \frac{X_i}{X_0} \;,\; V_0 = 1 \,,$$

wobei q_{ij} die mittlere Routingfrequenz von der i-ten zur j-ten Station, X_i den mittleren Durchsatz und V_i das mittlere Besuchsverhältnis der i-ten Station bezeichnet. Beide Gleichungssysteme drücken in äquivalenter Weise die Verbindungsstruktur (Topologie) des Systems innerhalb des Messzeitraumes aus. In Matrixnotation lautet das zweite Gleichungssystem

$$\boldsymbol{V} = \boldsymbol{V}\boldsymbol{Q} \Leftrightarrow (V_0, \ldots, V_M) = (V_0, \ldots, V_M) \begin{pmatrix} q_{00} & \cdots & q_{0M} \\ \vdots & \ddots & \vdots \\ q_{M0} & \cdots & q_{MM} \end{pmatrix} \;,\; V_0 = 1 \,.$$

Notwendige aber nicht hinreichende Bedingung für Jobflussgleichgewicht und operationale Verbundenheit ist, dass in der *Routingmatrix* $\boldsymbol{Q}$ jede Zeilensumme (Stationsausgang) gleich eins und jede Spaltensumme (Stationseingang) größer null ist.

Um den Umgang mit den Verkehrsgleichungen zu demonstrieren, diskutieren wir im folgenden vier einfache Beispiele.

Erstes Beispiel. Man betrachte den operational verbundenen Rechner in Abbildung 1.12 und nehme an, dass an ihm die mittleren Routingfrequenzen $q_{12} = 3/50$ und $q_{13} = 1/25$ im Jobflussgleichgewicht gemessen werden. Aufgrund der abgebil-

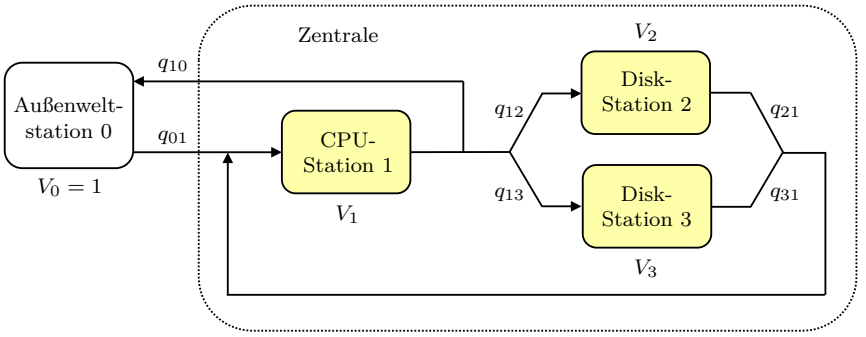


Abbildung 1.12. Rechner mit drei Bedienstationen plus seiner Außenweltstation 0. Eingetragen sind die Besuchsverhältnisse V_i sowie die relevanten Routingfrequenzen q_{ki}.

deten Richtungspfeile lassen sich die übrigen Routingfrequenzen leicht ermitteln, und man erhält für die gesuchten Besuchsverhältnisse die Verkehrsgleichungen

$$(1, V_1, V_2, V_3) = (1, V_1, V_2, V_3) \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{9}{10} & 0 & \frac{3}{50} & \frac{1}{25} \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} .$$

Offensichtlich erfüllt hierin die Routingmatrix die in Satz 1.10 zuletzt genannten Zeilen- und Spaltenbedingungen. Die spaltenweise Entwicklung der Matrixgleichung führt auf die vier Gleichungen

$$1 = \frac{9}{10} V_1 \ , \ V_1 = 1 + V_2 + V_3 \ , \ V_2 = \frac{3}{50} V_1 \ , \ V_3 = \frac{1}{25} V_1 \, ,$$

von denen genau eine redundant ist. Die zugehörige Lösung ist eindeutig und lautet

$$V_1 = \frac{10}{9} \ , \ V_2 = \frac{1}{15} \ , \ V_3 = \frac{2}{45} \, .$$

Unter der zusätzlichen Annahme, dass der mittlere Rechnerdurchsatz im Messzeitraum $X_0 = 90/\mathrm{s}$ beträgt, folgen nach dem Durchsatzgesetz die einzelnen Stationsdurchsätze

$$X_1 = V_1 X_0 = 100/\mathrm{s} \ , \ X_2 = V_2 X_0 = 6/\mathrm{s} \ , \ X_3 = V_3 X_0 = 4/\mathrm{s} \, .$$

Zweites Beispiel. Als nächstes gehe man davon aus, dass am selben Rechner lediglich die soeben berechneten Besuchsverhältnisse $V_1 = 10/9$ und $V_3 = 2/45$ gemessen werden. Ausgangspunkt für die nun gesuchten Größen sind die Verkehrsgleichungen

$$\left(1, \frac{10}{9}, V_2, \frac{2}{45}\right) = \left(1, \frac{10}{9}, V_2, \frac{2}{45}\right) \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 - q_{12} - q_{13} & 0 & q_{12} & q_{13} \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} ,$$

wobei auch hier die Richtungspfeile in Abbildung 1.12 sowie die Zeilen- und Spaltenbedingungen in Satz 1.10 berücksichtigt sind. Wie zuvor haben wir es mit drei linear unabhängigen Gleichungen für drei Unbekannte zu tun, in diesem Fall für die beiden Routingfrequenzen q_{12}, q_{13} und das Besuchsverhältnis V_2. Beschränken wir uns der Einfachheit halber auf die letzten drei Gleichungen,

$$\frac{10}{9} = 1 + V_2 + \frac{2}{45}\ ,\ V_2 = \frac{10}{9} q_{12}\ ,\ \frac{2}{45} = \frac{10}{9} q_{13}\,,$$

so folgt daraus die eindeutige Lösung

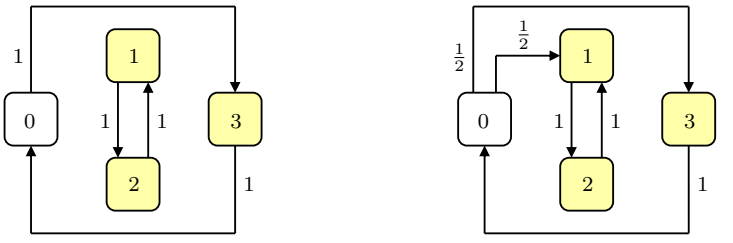
$$V_2 = \frac{1}{15}\ ,\ q_{12} = \frac{3}{50}\ ,\ q_{13} = \frac{1}{25}\,.$$

Drittes Beispiel. Was ist davon zu halten, wenn an einem beliebigen Rechner im Jobflussgleichgewicht die Routingmatrix

$$\boldsymbol{Q} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

gemessen wird? Wie man anhand der nullten und dritten Zeile erkennt, wird jeder von außen kommende Job zu Station 3 und von dort wieder nach außen geleitet, so dass die Stationen 1 und 2 von keinem dieser Jobs besucht werden. Die gemessenen Einträge in der ersten und zweiten Zeile können daher nur von Jobs stammen, die sich bereits am Anfang der Messung im Rechner befinden und ausschließlich zwischen den Stationen 1 und 2 zirkulieren (siehe Abbildung 1.13-links). Obwohl also die Routingmatrix die Zeilen- und Spaltenbedingungen in Satz 1.10 erfüllt, ist hier das Prinzip der operationalen Verbundenheit verletzt, weil die Stationen 1 und 2 von den übrigen Stationen entkoppelt sind und einen eigenen geschlossenen Verbund bilden. Dementsprechend ist die Lösung der Verkehrsgleichungen nicht eindeutig:

$$(1, V_1, V_2, V_3) = (1, V_1, V_2, V_3) \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \Rightarrow V_3 = 1\ ,\ V_1 = V_2\,.$$



Drittes Beispiel Viertes Beispiel

Abbildung 1.13. Visualisierung der Routingmatrix des dritten und vierten Beispiels.

Die Seiten 21 bis 244 sind
nicht Bestandteil der Leseprobe

Kapitel 4

Leistungsmessungen mittels Performancetests

Nachdem wir uns in den vorangegangenen drei Kapiteln mit der *Prognose* des Leistungsverhaltens von IT-Systemen auf der Grundlage von operationalen Warteschlangenmodellen beschäftigt haben, wollen wir uns in diesem Kapitel mit einem komplementären Thema auseinandersetzen, nämlich mit der *Messung* des Leistungsverhaltens von IT-Systemen im Rahmen von Performancetests. Der gedankliche Leitfaden hierzu lautet wie folgt:

- Ein Performancetest ist ein physikalisches Experiment, in welchem ein IT-Zentralsystem mittels Lasttreiber in offener oder geschlossener Form über einen längeren Zeitraum hinweg künstlich unter Last gesetzt wird und die damit einhergehenden Systemreaktionen (Antwortzeiten, Durchsätze, Auslastungen etc.) fortlaufend gemessen werden. Das Testziel besteht in der quantitativen Bestimmung des Leistungsverhaltens der Zentrale und ihrer Server.
- Weil jeder Performancetest unkontrollierbaren und unvorhersehbaren Einflüssen unterliegt, ist es angebracht, (i) den Test als ein (abstraktes) dynamisches Zufallsexperiment, (ii) die mit dem Test verbundenen Messvorgänge als (abstrakte) stochastische Leistungsprozesse und (iii) die aus den Messvorgängen resultierenden Messreihen als (konkrete) Prozessrealisierungen aufzufassen.
- Vor diesem Hintergrund zielt das vorliegende Kapitel im wesentlichen auf die Beantwortung der folgenden Frage ab: *Gegeben sei ein Performancetest und ein damit verbundener Leistungsprozess. Wie lassen sich die interessanten Verteilungsparameter des Prozesses anhand von ein oder mehreren Prozessrealisierungen möglichst genau schätzen?*
- Hierbei spielt die statistische Schätztheorie eine entscheidende Rolle. Sie befasst sich mit der Schätzung der Verteilungsparameter einer Zufallsvariablen, und zwar auf der Grundlage von mehreren unabhängigen Variablenrealisierungen.

Anders als es die Kapitelüberschrift vielleicht vermuten lässt, sind wir also nicht an technischen, logistischen oder teleologischen Aspekten von Performancetests interessiert, sondern ausschließlich daran, wie sich die Messreihen von Performancetests – als Realisierungen von stochastischen Prozessen betrachtet – hinsichtlich der in Frage stehenden Verteilungsparameter statistisch valide analysieren lassen (→ statistische Analyse von Performancetests).

Im ersten Abschnitt diskutieren wir den Zufallscharakter von Performancetests, führen das zentrale Konzept des stochastischen Prozesses ein und skizzieren das prinzipielle Vorgehen bei der statistischen Analyse von Performancetests. Der zweite Abschnitt ist als eigenständiger Einschub organisiert und behandelt in mehreren Unterabschnitten die für unsere Zwecke benötigten Elemente der statistischen Schätztheorie. In Abschnitt 4.3 stellen wir eine spezielle Form von Performancetests (genauer: von Performancetestmodellen) vor. Ihre Besonderheit besteht darin, dass sie auf diskreten Ereignissimulationen beruhen und deshalb nicht mit den praktischen Problemen und Restriktionen von realen Performancetests behaftet sind. Dementsprechend eignen sie sich besonders gut, um die in der Folge entwickelten Analysemethoden anhand von konkreten Beispielen „in Reinkultur“ zu demonstrieren.

Nach all diesen Vorbereitungen beschäftigen sich die letzten beiden Abschnitte mit der eigentlichen statistischen Analyse von Performancetests, und zwar (wie bereits erwähnt) mit dem Ziel, bestimmte Verteilungsparameter eines Leistungsprozesses anhand von ein oder mehreren Prozessrealisierungen möglichst genau zu schätzen. In Abschnitt 4.4 geht es um den sogenannten befristeten Performancetest. Hierbei liegt der Fokus auf dem Erwartungswert einer im Vorfeld festgelegten Prozessvariablen, so dass der Test nach der Messung dieser Variablen beendet werden kann. Abschnitt 4.5 befasst sich mit dem unbefristeten Performancetest. Bei ihm gilt das Interesse dem stationären (langfristigen) Prozesserwartungswert, mit der Konsequenz, dass der Test kein natürliches Ende hat und möglichst lange laufen sollte. Weil der unbefristete Fall anspruchsvollere Betrachtungen und Analysestrategien erfordert, ist Abschnitt 4.5 ebenfalls in mehrere Unterabschnitte unterteilt. Der letzte Unterabschnitt ist insofern von besonderer Bedeutung, als dass er den Zusammenhang zwischen der operationalen und der stochastischen Sicht auf unbefristete Performancetests aufzeigt.

4.1 Stochastische Beschreibung von Performancetests

Zunächst sollten wir klären, was genau wir in diesem Kapitel unter einem Performancetest verstehen wollen:

Definition 4.1: Performancetest

Ein Performancetest ist ein physikalisches Experiment, in welchem ein IT-Zentralsystem (eine IT-Zentrale) mit Hilfe von Lasttreibern in offener oder geschlossener Form über einen längeren Zeitraum hinweg künstlich unter Last gesetzt wird und die diesbezüglichen Systemreaktionen (Antwortzeiten, Auslastungen, Durchsätze etc.) fortlaufend gemessen werden. Zusätzlich gilt:

- Der Performancetest selbst ist ein isoliertes IT-System, bestehend aus den Lasttreibern (Lastservern), der Zentrale und etwaigen Überwachungsservern, die alle netzwerktechnisch miteinander verbunden sind. Es findet keine Beeinflussung des Performancetests von außen statt.

- Die Lasterzeugung erfolgt in den Lasttreibern, und zwar mit Hilfe eines deterministischen oder pseudozufälligen (durch einen Zufallszahlengenerator gesteuerten) Lastalgorithmus.
- *Kontrollierbare Anfangsbedingungen.* Unmittelbar vor Testbeginn wird jeder Server (durch Neustart, Rücksetzung des Datenbestandes etc.) in seinen „makroskopischen“ Initialzustand versetzt. Der Test beginnt mit dem Start der Lasterzeugung, also mit einer „leeren“ Zentrale.
- Der Zweck des Tests besteht darin, das Leistungsverhalten der Zentrale und ihrer Server quantitativ zu bestimmen.

Angesichts der heutigen komplexen, verteilten und hochgradig virtualisierten IT-Infrastrukturen stellt diese Definition ein in vielen Fällen nicht erreichbares Idealbild dar. So ist es insbesondere oftmals nicht möglich, (i) alle Server eines Performancetests vor Testbeginn zu initialisieren und (ii) den Performancetest gegenüber seiner Außenwelt exakt abzugrenzen. Trotzdem halten wir an der Definition fest und greifen die Probleme (i) und (ii) am Ende des Abschnittes noch einmal auf.

Beispiel: Performancetestprojekt. Stellen wir uns nun als einführendes Beispiel einen einfachen geschlossenen Performancetest mit der folgenden Testplanung vor:

- Der Test besteht aus einem Lasttreiber und einer Zentrale mit einem Server. Überwachungsserver sind nicht im Einsatz.
- Der Lasttreiber simuliert 10 virtuelle User, die alle exakt dieselbe Anfrage (denselben Job) an die Zentrale senden, und zwar mit einer Denkzeit von einer Sekunde.
- Der Test dauert eine Stunde. Gemessen wird der mittlere Zentraldurchsatz in diesem Zeitraum. Die Messung ist fehlerfrei.

Obwohl die Testdynamik durch die Testplanung augenscheinlich komplett festgelegt ist (keine Variationen in der Lasterzeugung und im Jobrouting, jeder Server startet in seinem Initialzustand, keine Messfehler), werden n Testwiederholungen trotzdem n unterschiedliche Durchsätze liefern. Der Grund hierfür sind unkontrollierbare und unvorhersehbare hardware- und softwarebedingte Störungen, zum Beispiel in Form von thermischen, elektromagnetischen und quantenmechanischen Interferenzen, zeitlichen Schwankungen und Fehlern bei der Datenübertragung sowie nebenläufigen CPU-, Speicher- und I/O-Zugriffen, die allesamt zu erratischen Änderungen der Arbeitsabläufe in der Zentrale und im Lasttreiber führen. Hinzu kommt, dass der physikalisch-technische „mikroskopische“ Initialzustand der Server bei jeder Testwiederholung anders ist.[26]

In Verallgemeinerung dieser Tatsachen lässt sich deshalb feststellen, dass die Dynamik eines jeden Performancetests durch nicht-funktionale bzw. nicht-kausale Mechanismen beeinflusst wird. Hieraus wiederum folgt, dass jeder Performancetest seinem Wesen nach ein *dynamisches Zufallsexperiment* ist und dementsprechend mit stochastischen Mitteln beschrieben und analysiert werden sollte. Dies gilt na-

[26] Die Störungsursachen lassen sich abstrakt durch die Begriffe *Physik*, *Nebenläufigkeit* und *Emergenz* zusammenfassen. Emergenz bedeutet, dass sich das Verhalten eines hinreichend komplexen Systems nicht vollständig aus dem Verhalten seiner Teile erklären lässt.

türlich erst recht für Performancetests, in denen pseudozufällige und bei jeder Testwiederholung anders initialisierte Algorithmen für die Lasterzeugung (in den Lasttreibern) oder das Jobrouting (in der Zentrale) zum Einsatz kommen.

Stochastische Prozesse. Die mit dynamischen Zufallsexperimenten verbundenen Messvorgänge lassen sich als *stochastische Prozesse* auffassen, die wir in natürlicher Erweiterung der Definition A.5 einer Zufallsvariablen wie folgt definieren:

Definition 4.2: Stochastischer Prozess

Gegeben sei ein dynamisches Zufallsexperiment mit der Indexmenge I der betrachteten Zeitpunkte (oder Zeitabschnitte) und der Ergebnismenge Ω.

- Die Abbildung $Y\colon \Omega \times I \to W\ ,\ (\omega, t) \mapsto Y(\omega, t)$ heißt *stochastischer Prozess*, wobei W dessen Wertemenge bezeichnet.
- Für ein festes $\omega_0 \in \Omega$ heißt die Abbildung $Y_{\omega_0}\colon I \to W\ ,\ t \mapsto Y(\omega_0, t) = Y_{\omega_0}(t)$ *Realisierung*, *Trajektorie* oder *Pfad* von Y.
- Für ein festes $t_0 \in I$ ist die Abbildung $Y_{t_0}\colon \Omega \to W\ ,\ \omega \mapsto Y(\omega, t_0) = Y_{t_0}(\omega)$ eine normale Zufallsvariable.
- Je nach Beschaffenheit der Index- und der Wertemenge nennt man Y *zeitdiskret* oder *zeitstetig* und *wertediskret* oder *wertestetig*.
- Üblicherweise verwendet man für Y die Schreibweisen $\{Y(t), t \in I\}$ oder $\{Y(t)\}$ und insbesondere im zeitdiskreten Fall $\{Y_t, t \in I\}$ oder $\{Y_t\}$.

Demnach ist ein stochastischer Prozess eine zeitlich geordnete Folge von Zufallsvariablen, die in jeder Wiederholung eines dynamischen Zufallsexperimentes konkrete Werte annehmen. Diese Werte sind einerseits durch das zufällige Ergebnis ω der jeweiligen Wiederholung bestimmt und konstituieren andererseits die Prozessrealisierung in dieser Wiederholung. Dabei sind folgende Punkte besonders zu berücksichtigen:

- Die Werte derselben Prozessrealisierung hängen möglicherweise voneinander ab, während die Werte verschiedener Prozessrealisierungen in jedem Fall unabhängig voneinander sind.
- Das Ergebnis ω kann man sich als eine komplexe, hochdimensionale Funktion der Zeit, $\omega(t)$, vorstellen, die den momentanen Zustand des Zufallsexperimentes in der betreffenden Wiederholung beschreibt.
- Jede Wiederholung findet unter denselben *kontrollierbaren* Anfangsbedingungen statt. Der Zufallscharakter der Wiederholungen resultiert zum einen aus diversen *unkontrollierbaren* Anfangsbedingungen und zum anderen aus permanenten *unkontrollierbaren* Störungen, die allesamt echt zufällig oder aus Mangel an Informationen als zufällig anzusehen sind.

Um all dies zu veranschaulichen, kommen wir zu unserem obigen Performancetestprojekt zurück und betrachten den zeitstetigen und wertediskreten stochastischen Prozess $\{N(t), t \in \mathbb{R}_0^+\}$ der momentanen Zentralbesetzung, wobei t die Testzeit bedeutet. Abbildung 4.1 zeigt drei mögliche Realisierungen $N(\omega_1, t)$, $N(\omega_2, t)$ und $N(\omega_3, t)$ des Prozesses in den ersten drei Testwiederholungen. Alle drei Realisierungen starten unter denselben kontrollierbaren Anfangsbedingungen (vorletzter

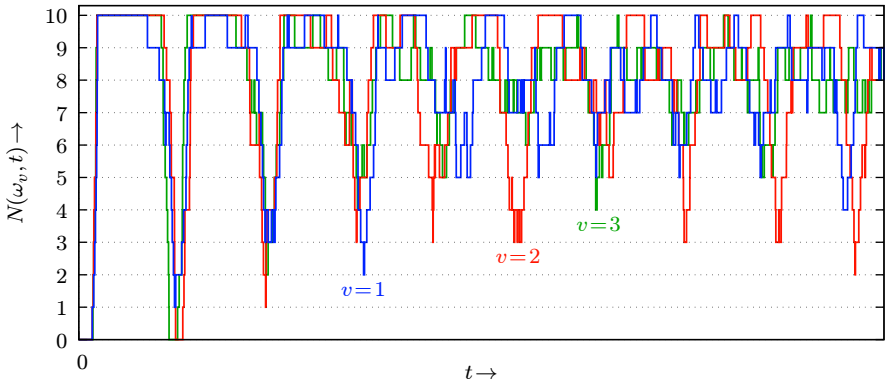


Abbildung 4.1. Drei mögliche Realisierungen des zum obigen Performancetestprojekt gehörenden Zentralbesetzungsprozesses $\{N(t)\}$ in Abhängigkeit von der Testzeit. Jede Prozessrealisierung v startet mit dem Wert $N(\omega_v, 0) = 0$ und nimmt einen unvorhersehbaren Verlauf. Der Prozess selbst startet mit der „scharfen" Wahrscheinlichkeitsfunktion $\mathbb{P}\{N(0) = n\} = \left\{ \begin{matrix} 1 \ , \ n = 0 \\ 0 \ , \ \text{sonst} \end{matrix} \right\}$, die im Laufe der Zeit über alle möglichen Prozesswerte $n \in [0 : 10]$ „verschmiert".

Punkt in Definition 4.1) und laufen aus den genannten Gründen schon recht bald auseinander. Die Realisierungen würden noch schneller auseinanderlaufen, wenn man zum Beispiel die einzelnen Denkzeiten mit Hilfe eines Zufallszahlengenerators variabilisieren und den Generator bei jeder Testwiederholung anders initialisieren würde. Die Besetzungszahlen derselben Prozessrealisierung hängen voneinander ab, weil sich die Zentrale aus funktionalen Gründen nicht spontan füllen oder leeren kann. Die Besetzungszahlen verschiedener Realisierungen sind dagegen unabhängig voneinander, weil verschiedene Testwiederholungen nicht miteinander kommunizieren und daher auch in keinem kausalen Zusammenhang stehen.

Leistungsprozesse von Performancetests. Im allgemeinen besitzen die experimentell relevanten Prozesse von Performancetests eine andere Struktur als $\{N(t)\}$. Sie sind allesamt zeitdiskret, meistens wertestetig und von der Art

$$\{Y_k, k \in \mathbb{N}\} \ , \ Y_k \equiv \left\{ \begin{matrix} k\text{-te Messung einer} \\ \text{Leistungsgröße} \end{matrix} \right\} \quad \textit{(Leistungsprozess)} \, , \tag{4.1}$$

wobei sich jede Messung auf ein eigenes Testzeitintervall (gegebenenfalls der Länge null) bezieht, das entweder von vornherein oder durch das Eintreten von bestimmten Zufallsereignissen festgelegt ist. Beispiele für Y_k sind

- die k-te stationsseitige mittlere Jobverweilzeit, bezogen auf das feste Testzeitintervall $\Delta T_k = [(k-1)\Delta T : k\Delta T]$,
- die k-te stationsseitige Jobverweilzeit $\equiv$ Testzeitintervall zwischen dem zufälligen Eintritts- und Austrittszeitpunkt des k-ten ausgetretenen Jobs,
- die k-te momentane Stationsbesetzung, bezogen auf den zufälligen Eintrittszeitpunkt des k-ten eintretenden Jobs.

Wie man sich leicht klarmacht, gilt auch bei den meisten dieser sogenannten *Leistungsprozesse*, dass die Werte derselben Prozessrealisierung voneinander abhängen.

Asymptotisch stationäre und ergodische Prozesse. Wichtige Kenngrößen eines Prozesses $\{Y_k\}$ sind seine Wahrscheinlichkeitsverteilungen $\mathbb{P}\{Y_k \leq y\}$, Erwartungswerte $\mathbb{E}[Y_k]$, Varianzen $\mathbb{V}[Y_k]$ und Kovarianzen $\mathrm{Cov}[Y_k, Y_l]$. Damit verbunden wollen wir nun spezielle und für unsere Zwecke nützliche Prozesskategorien einführen, die auf den bekannten Konzepten der *Stationarität* und *Ergodizität* beruhen:[27]

Definition 4.3: Stationärer, asymptotisch stationärer und ergodischer Prozess

Gegeben sei ein stochastischer Prozess $\{Y_k, k \in \mathbb{N}\}$.

- Der Prozess heißt *stationär*, wenn für alle $k, s \in \mathbb{N} \wedge y \in W$

$$\mathbb{P}\{Y_k \leq y\} = F(y)\ ,\ \mathbb{E}[Y_k] = \mu\ ,\ \mathbb{V}[Y_k] = \sigma^2\ ,\ \mathrm{Cov}[Y_k, Y_{k+s}] = \gamma(s)\,,$$

  wobei $\gamma(s)$ *Autokovarianzfunktion* genannt wird.
- Der Prozess heißt *asymptotisch stationär*, wenn für alle $s \in \mathbb{N} \wedge y \in W$

$$\lim_{k\to\infty} \mathbb{P}\{Y_k \leq y\} = F(y)\ ,\ \lim_{k\to\infty} \mathbb{E}[Y_k] = \mu\ ,\ \lim_{k\to\infty} \mathbb{V}[Y_k] = \sigma^2$$
$$\lim_{k\to\infty} \mathrm{Cov}[Y_k, Y_{k+s}] = \gamma(s)\,.$$

- Der Prozess heißt *ergodisch*, wenn er wenigstens asymptotisch stationär ist und darüber hinaus die folgende Bedingung erfüllt:

$$\lim_{n\to\infty} \mathbb{E}\left[\left(\frac{1}{n}\sum_{k=1}^{n} Y_k - \mu\right)^2\right] = 0\,.$$

Stationarität bedeutet also, dass die Verteilung und somit auch der Erwartungswert und die Varianz des Prozesses über alle Zeitpunkte (Zeitindizes) k hinweg konstant sind. Zusätzlich ist die Abhängigkeitsstruktur innerhalb des Prozesses dadurch charakterisiert, dass die Kovarianz von zwei Prozessvariablen allein von derem zeitlichen Abstand (und nicht von deren zeitlichen Absolutpositionen) abhängt. Asymptotische Stationarität bringt zum Ausdruck, dass sich der Prozess im Laufe der Zeit zu einem stationären Prozess hinentwickelt. Dabei verläuft die Entwicklung dergestalt, dass der Prozess schon ab einem bestimmten endlichen Zeitpunkt n_0 praktisch (im Rahmen der gegebenen Genauigkeit) nicht mehr von seiner stationären Version zu unterscheiden ist:

$$\mathbb{P}\{Y_k \leq y\} \approx F(y)\ ,\ \mathrm{Cov}[Y_k, Y_{k+s}] \approx \gamma(s)\ \forall\, k, s \in \mathbb{N} \wedge y \in W \wedge k \geq n_0\,.$$

Der letzte Punkt in der Definition ist äquivalent mit [siehe Aufgabe 104(a)]

$$\lim_{n\to\infty} \mathbb{V}\left[\frac{1}{n}\sum_{k=1}^{n} Y_k\right] = 0\,.$$

[27] Die Definition 4.3 weicht vom mathematischen Standard ab. Der Begriff „Ergodizität“ wurde ursprünglich von dem Physiker Ludwig Boltzmann im Rahmen der statistischen Mechanik geprägt und setzt sich aus den griechischen Wörtern für Energie und Weg zusammen.

Somit besagt Ergodizität im wesentlichen, dass der zeitliche Mittelwert einer jeden einzelnen Prozessrealisierung im Laufe der Zeit bzw. mit zunehmender Zahl der realisierten Werte gegen den stationären Erwartungswert μ konvergiert [siehe Aufgabe 104(b)]:

$$\lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} Y_k(\omega) = \mu \;\forall\, \omega \in \Omega \quad \textit{(Zeitmittel=Ensemblemittel)}\,. \tag{4.2}$$

Dabei ist besonders der Umstand zu würdigen, dass im Regelfall zu jeder einzelnen Prozessrealisierung ein anderer Initialzustand $\omega(t = 0)$ des Zufallsexperimentes gehört.

Was den Leistungsprozess (4.1) im Rahmen von Performancetests betrifft, so ist aufgrund des vorletzten Punktes in Definition 4.1 klar, dass er mit Sicherheit nicht stationär ist. In vielen Fällen besteht jedoch die berechtigte Hoffnung, dass der Prozess asymptotisch stationär und darüber hinaus auch ergodisch ist. Beide Eigenschaften zusammen genommen sind eine mathematische Umschreibung dessen, was landläufig als „Einschwingvorgang“ von Performancetests bezeichnet wird. Allerdings ist im Auge zu behalten, dass nicht ein einzelner Test bzw. eine einzelne Prozessrealisierung „einschwingt“, sondern allenfalls die unendliche Gesamtheit aller (gedanklich parallelen) Testwiederholungen bzw. der betreffende Prozess selbst.

Stichprobenprozesse und Stichprobenvariablen. Eine erweiterte und für die statistische Analyse günstige Anschauungsweise des Leistungsprozesses (4.1) besteht darin, ihn in Gedanken unendlich oft zu kopieren und jeder Wiederholung v des Performancetests eine eigene Prozesskopie $\{Y_k^{(v)}\}$ zuzuordnen, die genau eine Realisierung besitzt, nämlich jene der v-ten Wiederholung. In Anlehnung an die Terminologie der Statistik nennen wir die unabhängigen Prozesskopien $\{Y_k^{(v)}\}$ *Stichprobenprozesse* von $\{Y_k\}$ und die unabhängigen Zufallsvariablenkopien $Y_k^{(v)}$ *Stichprobenvariablen* von Y_k.

Statistische Analyse von Performancetests. Nach all diesen vorbereitenden Überlegungen sind wir jetzt in der Lage, das Problem der statistischen Analyse von Performancetests genauer zu umreißen. Zu diesem Zweck stellen wir uns einen Performancetest und einen Leistungsprozess $\{Y_k\}$ vor, wobei (i) der Test m mal ausgeführt bzw. wiederholt wird, (ii) jede Wiederholung n Leistungsmessungen umfasst und (iii) jede Wiederholung unter denselben kontrollierbaren Anfangsbedingungen gemäß Definition 4.1 startet. Die zugehörige stochastische Situation lässt sich dann in der Form

$$\left(\begin{array}{cccc|ccc} Y_1^{(1)} & \dots & Y_b^{(1)} & \dots & Y_{n_0}^{(1)} & \dots & Y_n^{(1)} \\ Y_1^{(2)} & \dots & Y_b^{(2)} & \dots & Y_{n_0}^{(2)} & \dots & Y_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Y_1^{(m)} & \dots & Y_b^{(m)} & \dots & Y_{n_0}^{(m)} & \dots & Y_n^{(m)} \end{array}\right) \longrightarrow \left(\begin{array}{cccc|ccc} y_1^{(1)} & \dots & y_b^{(1)} & \dots & y_{n_0}^{(1)} & \dots & y_n^{(1)} \\ y_1^{(2)} & \dots & y_b^{(2)} & \dots & y_{n_0}^{(2)} & \dots & y_n^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y_1^{(m)} & \dots & y_b^{(m)} & \dots & y_{n_0}^{(m)} & \dots & y_n^{(m)} \end{array}\right)$$

zusammenfassen, und zwar mit der folgenden Interpretation: Die linke Seite repräsentiert die *wahrscheinlichkeitstheoretische Sicht* auf den Performancetest *vor* seinen Ausführungen. Jede Zeile steht für eine Testwiederholung und enthält einen stochastisch unabhängigen Stichprobenprozess. Jeder Stichprobenprozess wiederum besteht aus n stochastisch abhängigen Stichprobenvariablen. Die Stichprobenvariablen derselben Spalte sind nicht nur stochastisch unabhängig sondern auch identisch verteilt. Der Zeitraum links vom Trennstrich heißt *transiente Phase*. Dort sind die Stichprobenprozesse nicht-stationär. Unter der Voraussetzung, dass $\{Y_k\}$ asymptotisch stationär ist, haben wir rechts vom Trennstrich die *stationäre Phase*, wo die Stichprobenprozesse in guter Näherung als stationär angesehen werden können. A priori sind die Verteilungen und Verteilungsparameter (Erwartungswerte, Varianzen, Kovarianzen etc.) der Stichprobenprozesse sowie die Länge der transienten Phase unbekannt.

Die rechte Seite symbolisiert die *statistische Sicht* auf den Performancetest *nach* all seinen Ausführungen und enthält in derselben Darstellungslogik die realisierten bzw. gemessenen Werte *(Stichprobenwerte)* der einzelnen Stichprobenvariablen. Die Hauptaufgabe der statistischen Analyse besteht nun darin, allein auf der Basis der endlich vielen Messwerte die interessierenden Verteilungsparameter des Leistungsprozesses $\{Y_k\}$ möglichst akkurat zu schätzen. Dabei lassen sich grundsätzlich zwei Interessenschwerpunkte identifizieren, mit denen verschiedene Typen von Performancetests verbunden sind, nämlich *befristete* und *unbefristete Tests*.

Befristete Performancetests. Bei dieser Art von Tests ist man an dem Erwartungswert einer bestimmten Prozessvariablen Y_b von $\{Y_k\}$ interessiert, so dass die v-te Testwiederholung beim Eintreten des Ereignisses {Messung von $Y_b^{(v)}$ ist abgeschlossen} beendet werden kann. Die dazu passende statistische Analyse beruht vor allem auf der Tatsache, dass die Stichprobenvariablen $Y_b^{(v)}$ bei festem b unabhängig und identisch verteilt sind. Das Ergebnis der Analyse sind ein Schätzwert und ein damit verbundenes Schätzintervall für $\mathbb{E}[Y_b]$. Beide Größen tragen der intuitiven Erwartung Rechnung, dass $\mathbb{E}[Y_b]$ umso genauer geschätzt werden kann, je mehr Stichprobenwerte $y_b^{(v)}$ vorliegen.

Unbefristete Performancetests. Hier liegt das Augenmerk auf dem stationären (langfristigen) Erwartungswert $\lim_{k\to\infty} \mathbb{E}[Y_k]$ des Leistungsprozesses $\{Y_k\}$, was natürlich zur Voraussetzung hat, dass $\{Y_k\}$ asymptotisch stationär ist. Dementsprechend gibt es hier kein ereignisinduziertes Testende, und jede Testwiederholung sollte „hinreichend lange“ laufen. Eine Analysestrategie ist an den befristeten Fall angelehnt und besteht darin, sich auf die unabhängig und identisch verteilten Mittelwerte $\bar{Y}^{(v)}$ der Stichprobenvariablen $Y_{n_0}^{(v)}$, $Y_{n_0+1}^{(v)}$, $Y_{n_0+2}^{(v)}, \ldots$ zu konzentrieren und mit Hilfe der Realisierungen $\bar{y}^{(v)}$ eine Schätzung von $\mathbb{E}\left[\bar{Y}\right]$ vorzunehmen. Eine andere Analysestrategie richtet sich allein auf die Stichprobenvariablen $Y_{n_0}^{(1)}$, $Y_{n_0+1}^{(1)}$, $Y_{n_0+2}^{(1)}, \ldots$ des ersten Stichprobenprozesses aus. Dieser Ansatz macht allerdings nur dann Sinn, wenn der Prozess $\{Y_k\}$ ergodisch ist, weil nur dann die statistische Schätzgenauigkeit (wie im befristeten Fall) mit der Zahl der Stich-

probenwerte wächst. Beide Strategien sind mit dem Problem behaftet, dass der Startzeitpunkt n_0 der stationären Phase zunächst unbekannt ist *(Transienz- oder Einschwingproblem)*. Bei der zweiten Strategie kommt hinzu, dass die Stichprobenvariablen $Y_{n_0}^{(1)}, Y_{n_0+1}^{(1)}, Y_{n_0+2}^{(1)}, \ldots$ zwar nahezu identisch verteilt aber meistens nicht unabhängig voneinander sind *(Autokorrelationsproblem)*.

Mit all diesen Aspekten werden wir uns in den Abschnitten 4.3 bis 4.5 näher beschäftigen, nachdem im folgenden Abschnitt die dazu notwendigen statistischen Grundlagen geschaffen wurden. Jetzt kommen wir noch einmal ganz konkret auf den Begriff der Ergodizität zurück sowie auf den Zusammenhang von Zufallsexperimenten, Performancetests, Leistungsprozessen und Leistungsmessungen.

Ergodizität am Beispiel von Münzwurfserien. Um die Eigenschaft der Ergodizität noch besser zu verstehen, betrachte man die beiden zeit- und wertediskreten Prozesse

$$\{U_k, k \in \mathbb{N}\}\ ,\ U_k = \left\{ \begin{array}{l} 0\ ,\ k\text{-ter Wurf} = \text{Zahl} \\ 1\ ,\ k\text{-ter Wurf} = \text{Kopf} \end{array} \right\}$$

$$\{G_k, k \in \mathbb{N}\}\ ,\ G_k = \left\{ \begin{array}{ll} 0 & ,\ k \leq 2 \wedge k\text{-ter Wurf} = \text{Zahl} \\ 1 & ,\ k \leq 2 \wedge k\text{-ter Wurf} = \text{Kopf} \\ G_2 & ,\ k > 2 \end{array} \right\},$$

mit

$$\mathbb{P}\{U_k = 0\} = \mathbb{P}\{U_k = 1\} = \mathbb{P}\{G_k = 0\} = \mathbb{P}\{G_k = 1\} = \mathbb{E}[U_\infty] = \mathbb{E}[G_\infty] = \frac{1}{2}\,.$$

Der erste Prozess beschreibt eine Wurfserie mit einer normalen ungezinkten Münze. Beim zweiten Prozess wird eine gezinkte Münze geworfen, die kurioserweise nach dem zweiten Wurf „einfriert“ und in allen Folgewürfen dieselbe Seite zeigt wie beim zweiten Wurf. In Aufgabe 106 wird auf der Basis von Definition 4.3 formal gezeigt, dass $\{U_k\}$ stationär und ergodisch ist, während $\{G_k\}$ zwar asymptotisch stationär (ab $k = 2$ sogar stationär) aber nicht ergodisch ist. Die Unterschiede der beiden Prozesse bezüglich Ergodizität lassen sich aber auch heuristisch über die Äquivalenz [siehe (4.2)]

$$\text{Ergodizität} \Leftrightarrow \lim_{n\to\infty} \frac{1}{n} \sum_{k=1}^{n} Y_k(\omega) = \mu = \mathbb{E}[Y_\infty]\ \forall\, \omega \in \Omega$$

begründen: Jede Realisierung von $\{U_k\}$ ist von der Form

$$\{(0|1), (0|1), (0|1), \ldots\} \quad [(0|1) \equiv \text{„0 oder 1“}]\,,$$

wobei alle Werte unabhängig voneinander sind. Somit kommen in allen Realisierungen die 0 und die 1 gleich oft vor, so dass die zeitlichen Mittelwerte aller Realisierungen dem Erwartungswert $\mathbb{E}[U_\infty]$ entsprechen. Jede Realisierung von $\{G_k\}$ besitzt dagegen die Form

$$\{(0|1), 0, 0, 0, \ldots\} \quad \text{oder} \quad \{(0|1), 1, 1, 1, \ldots\}\,,$$

und zwar mit dem zeitlichen Mittelwert 0 oder 1, der in keinem Fall mit $\mathbb{E}[G_\infty]$ übereinstimmt. Der hier eigentlich entscheidende Punkt ist, dass innerhalb einer Realisierung die Werte ab $k = 2$ zu stark aneinander gekoppelt sind, und diese Kopplung wiederum hängt mit der Autokovarianzfunktion $\gamma(s)$ zusammen [siehe Aufgabe 104(c)]. Der Prozess $\{G_k\}$ ist insofern ein Extrembeispiel für Nicht-Ergodizität, als dass er tatsächlich keine einzige Realisierung enthält, deren zeitlicher Mittelwert im Laufe der Zeit gegen $\mathbb{E}[G_\infty]$ konvergiert.

Performancetests als dynamische Zufallsexperimente. Abschließend diskutieren wir noch einen wichtigen Aspekt, der zum einen mit der Gleichsetzung Performancetest = dynamisches Zufallsexperiment (hier sollte man sich insbesondere den letzten Punkt der Bemerkungen zu Definition 4.2 in Erinnerung rufen) und zum anderen mit der endlichen Laufzeit von realen Tests zu tun hat. Zu diesem Zweck bemühen wir ein letztes mal unser anfängliches Performancetestprojekt und betrachten den Zentralantwortzeitprozess

$$\{R_k, k \in \mathbb{N}\}\ ,\ R_k \equiv \text{Verweilzeit des } k\text{-ten ausgetretenen Zentraljobs}\,, \tag{4.3}$$

dessen langfristiger Erwartungswert $\mu = \lim_{k\to\infty} \mathbb{E}[R_k]$ mit Hilfe eines einzigen einstündigen Tests und somit anhand einer einzigen endlichen Messreihe bzw. Prozessrealisierung ermittelt werden soll. Die Frage, der wir nachgehen wollen, lautet: *Mit welchen Problemen ist bei der Bestimmung von μ zu rechnen, wenn der Performancetest einige Voraussetzungen in Definition 4.1 nicht erfüllt?* Hierzu gehen wir nun die drei relevantesten Fälle der Reihe nach durch.

Erster Fall: Der Test erfüllt alle Voraussetzungen in Definition 4.1. In diesem Fall besteht unter milden Zusatzbedingungen eine Berechtigung zu der Annahme, dass der Prozess $\{R_k\}$ asymptotisch stationär und ergodisch ist und die Mittelwerte aller möglichen Messreihen mit zunehmender Zahl der Messwerte gegen μ konvergieren. Die Messreihe des einen durchgeführten Tests sollte in etwa (und stark vereinfacht) so aussehen wie in Abbildung 4.2, wobei der graue rechte Bereich die Erwartungs-

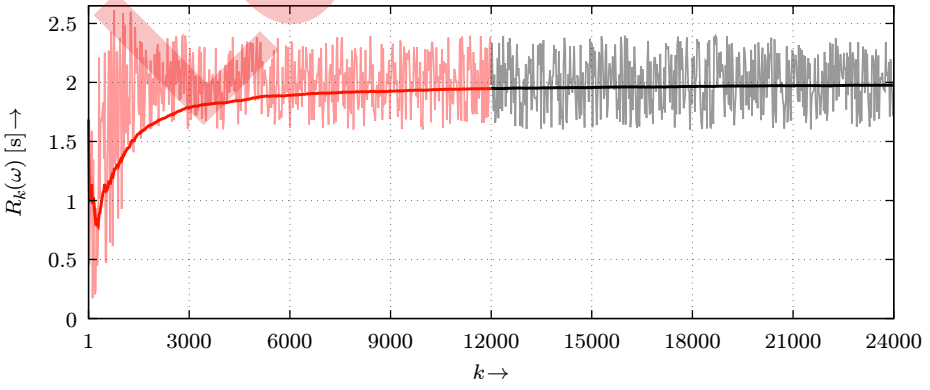


Abbildung 4.2. Mögliche Realisierung des Zentralantwortzeitprozesses (4.3) unter den in Definition 4.1 genannten Voraussetzungen. Die Zitterkurve repräsentiert die einzelnen Jobverweilzeiten in der Zentrale in Abhängigkeit vom Zeitindex k. Die fette Kurve ist der kumulative Mittelwert der Jobverweilzeiten.

haltung bei fiktiver Verlängerung der Testzeit widerspiegelt. Ob $\{R_k\}$ tatsächlich ergodisch ist, lässt sich natürlich nur anhand von mehreren Testwiederholungen empirisch prüfen. Ist Ergodizität mit hinreichender Sicherheit gegeben, dann bietet sich zur Bestimmung von μ die weiter oben genannte zweite Analysestrategie von unbefristeten Performancetests an. Ist Ergodizität nicht gegeben, dann ist der eine Test (mit Blick auf die Bestimmung von μ) wertlos.

Für die Werthaltigkeit des Tests ist allerdings die Ergodizität von $\{R_k\}$ nicht ausreichend. Zusätzlich muss wegen der Testzeitbegrenzung der Messreihenmittelwert *hinreichend schnell* gegen μ konvergieren (siehe die fette Kurve in Abbildung 4.2). Dies wiederum kann nur geschehen, wenn (i) innerhalb der Testzeit viele Besetzungsänderungen in der Zentrale stattfinden und (ii) relativ zur Testzeit alle unkontrollierbaren Störungen ausschließlich hochfrequente Anteile enthalten (siehe die Zitterkurve in Abbildung 4.2).

Zweiter Fall: Nicht alle Server werden vor Testbeginn initialisiert. Hier haben wir es schlicht mit einem anderen Performancetest bzw. Zufallsexperiment zu tun, bei dem (aus Mangel an Informationen) die mikroskopischen *und* makroskopischen Anfangszustände der uninitialisierten Server zu den unkontrollierbaren Anfangsbedingungen gehören. Dementsprechend ändert sich auch der Zentralantwortzeitprozess $\{R_k\}$, und zwar voraussichtlich dergestalt, dass er immer noch asymptotisch stationär aber nicht mehr ergodisch ist. Vermutlich wird man bei einem Test eine Messreihe wie in Abbildung 4.2 und bei einem anderen Test eine Messreihe wie in Abbildung 4.3 erhalten, die beide ähnlich aussehen, sich aber auf unterschiedlichen Niveaus stabilisieren, je nachdem, welchen funktionalen Einfluss die uninitialisierten Server auf den betreffenden Test haben.

Insgesamt folgt hieraus: *Werden vor Beginn des Performancetests nicht alle Server initialisiert, dann ist ein einzelner Test (mit Blick auf die Bestimmung des Erwartungswertes μ) ohne ein genaueres Verständnis der Effekte der Nicht-Initialisierungen wertlos.*

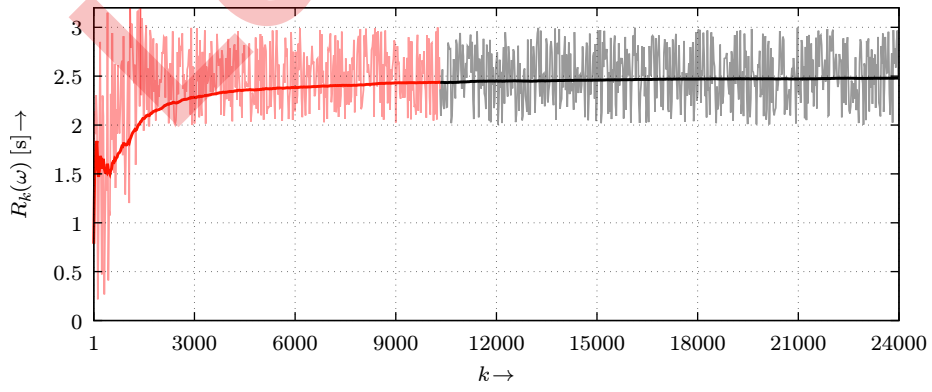


Abbildung 4.3. Mögliche Realisierung des Zentralantwortzeitprozesses (4.3) unter eingeschränkten kontrollierbaren Testanfangsbedingungen. Die Zitterkurve repräsentiert die einzelnen Jobverweilzeiten in der Zentrale in Abhängigkeit vom Zeitindex k. Die fette Kurve ist der kumulative Mittelwert der Jobverweilzeiten.

Dritter Fall: Der Test ist gegenüber seiner Außenwelt nicht exakt abgegrenzt. Dies bedeutet offensichtlich, dass der Test während seines Verlaufes möglicherweise unvorhersehbaren Fremdeinflüssen ausgesetzt ist. Deshalb müssen wir auch hier wieder von einem anderen Performancetest bzw. Zufallsexperiment ausgehen, bei dem nun (aus Mangel an Informationen) die Fremdeinflüsse den unkontrollierbaren Störungen zugerechnet werden. Abbildung 4.4 zeigt eine mögliche Messreihe des Tests inklusive ihres weiteren Verlaufes bei fiktiver Testzeitverlängerung. Die signifikanten

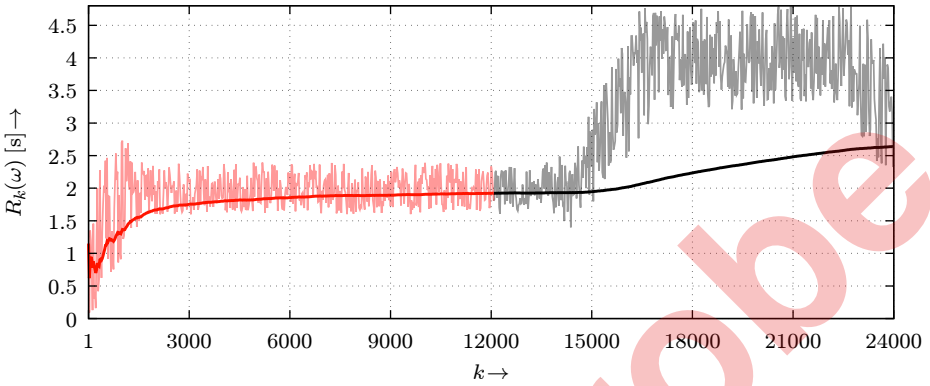


Abbildung 4.4. Mögliche Realisierung des Zentralantwortzeitprozesses (4.3) unter dem Einfluss von hoch- und niederfrequenten Störungen. Die Zitterkurve repräsentiert die einzelnen Jobverweilzeiten in der Zentrale in Abhängigkeit vom Zeitindex k. Die fette Kurve ist der kumulative Mittelwert der Jobverweilzeiten.

Abweichungen der Datenpunkte nach oben sollen die Effekte von Fremdeinflüssen – beispielsweise von sporadischen Fremdbelastungen des Netzwerkes – verdeutlichen. Weil die eigentliche Messreihe auf eine Stunde begrenzt ist, wird man in diesem Beispiel ohne das Wissen um die Fremdeinflüsse vermutlich zu derselben Einschätzung wie im ersten Fall kommen, nämlich dass der Zentralantwortzeitprozess $\{R_k\}$ ergodisch ist und der Mittelwert der Messreihe recht schnell gegen μ konvergiert. Tatsächlich verhält es sich aber so, dass (i) die Ergodizität des Prozesses komplett in Frage steht und (ii) aufgrund der niedrigen Frequenzen der Fremdeinflüsse (in der Größenordnung 1/h) eine echte Konvergenz des Messreihenmittelwertes frühestens – wenn überhaupt – nach vielen Tagen oder Wochen zu erwarten ist.

Mit anderen Worten: *Ist der Performancetest gegenüber seiner Außenwelt nicht exakt abgegrenzt, dann ist ein einzelner Test (mit Blick auf die Bestimmung des Erwartungswertes μ, falls er überhaupt existiert) ohne ein genaueres Verständnis der möglichen Fremdeinflüsse wertlos.*

Schauen wir uns der Vollständigkeit halber noch kurz den zu (4.3) „orthogonalen“ Zentralantwortzeitprozess (Stichwort: befristete Performancetests)

$$\left\{R_b^{(v)}, v \in \mathbb{N}\right\}\ ,\ v \equiv \text{Testwiederholung}\ ,\ b \text{ fest}$$

an. Losgelöst von der konkreten Zufallsbeschaffenheit des Performancetests ist dieser Prozess eine Serie von unabhängigen und identisch verteilten Zufallsvariablen

und somit in jedem Fall stationär und ergodisch [siehe Aufgabe 105(a)]. Daher konvergiert der Mittelwert einer jeden (jetzt aus vielen Testwiederholungen resultierenden) Messreihe immer gegen den Erwartungswert $\mathbb{E}\left[R_b^{(1)}\right]$. Allerdings ist hier zu berücksichtigen, dass sowohl uninitialisierte Server (zweiter Fall) als auch unvorhersehbare Fremdeinflüsse (dritter Fall) die Prozessstruktur und den Erwartungswert ändern und die Konvergenz des Messreihenmittelwertes verlangsamen.

4.2 Statistische Schätztheorie

Wie sich später zeigen wird, beruht die statistische Analyse von Performancetests zu einem Großteil auf der Beantwortung der folgenden Frage: *Gegeben sei eine Stichprobe $\{X_1, \ldots, X_n\}$ (also eine Folge von unabhängigen Kopien) der Zufallsvariablen X. Wie lassen sich der Erwartungswert $\mu = \mathbb{E}[X]$ und die Varianz $\sigma^2 = \mathbb{V}[X]$ allein auf der Basis der Stichprobenwerte $\{x_1, \ldots, x_n\}$ schätzen, und wie vertrauenswürdig sind diese Schätzungen?* Im vorliegenden Abschnitt wollen wir uns deshalb mit genau dieser Frage schwerpunktmäßig auseinandersetzen. Im ersten Unterabschnitt entwickeln wir zufällige Punktschätzer von μ und σ^2. Die nächsten beiden Unterabschnitte beschäftigen sich mit den Wahrscheinlichkeitsverteilungen der Punktschätzer. Daraus werden in den Unterabschnitten 4.2.4 und 4.2.5 zufällige Intervallschätzer abgeleitet, die μ und σ^2 mit einer definierten Wahrscheinlichkeit überdecken. Der letzte Unterabschnitt zeigt auf, wie sich bestimmte Parameterhypothesen – zum Beispiel der Form $\mu \neq \mu_0$, $\mu > \mu_0$ oder $\mu < \mu_0$ – auf der Basis von Intervallschätzungen statistisch valide prüfen lassen.

4.2.1 Punktschätzungen

Ein grundlegendes Konzept der statistischen Schätztheorie ist der *Punktschätzer*. Seine Definition lautet wie folgt:

Definition 4.4: Punktschätzer
Sei $\{X_1, \ldots, X_n\}$ eine Stichprobe der Zufallsvariablen X, die den Verteilungsparameter θ besitzt. Dann heißt die Zufallsvariable $\mathcal{E}_\theta(n) = h(X_1, \ldots, X_n)$ *Punktschätzer von* θ, wenn ihre Realisierung $e_\theta(n) = h(x_1, \ldots, x_n)$ als Schätzwert für θ verwendet wird. Wichtige Gütekriterien eines Schätzers sind:

- *Erwartungstreue.* Sein Erwartungswert liefert den wahren Wert:

$$\mathbb{E}[\mathcal{E}_\theta(n)] = \theta\,.$$

- *Asymptotische Erwartungstreue.* Sein Erwartungswert konvergiert gegen den wahren Wert:

$$\lim_{n\to\infty} \mathbb{E}[\mathcal{E}_\theta(n)] = \theta\,.$$

Die Seiten 258 bis 438 sind
nicht Bestandteil der Leseprobe

Sachverzeichnis