Entwicklung optimaler Zuteilungsstrategien für Rechner-Modelle durch Simulation und Rechnung*

Development of optimal strategies for computer-models by simulation and computation

Elektron. Rechenanl. 16 (1974), H. I, S. 9-17
Manuskripteingang: 17. 5. 1973

von B. WALKE
Bereich Forschung und Entwicklung der AEG-Telefunken, Ulm/Donau

Die zwei Hauptaufgaben der Betriebsmittelverwaltung sind Rechnerkern- und Arbeitsspeicherzuteilung. Es wird für ein Teilnehmerrechensystem gezeigt, wie man Simulation und Rechnung einsetzen kann, um berechenbare wirklichkeitsnahe Modelle zu entwickeln. Die Programmbearbeitung ist eine Folge von Transport- und Rechenzeiten für "Teilaufgaben". Von großer Bedeutung ist die statistische Verteilung der Rechenzeiten von Teilaufgaben. Meßkurven werden durch hyper-, stückweise- und entartet-negativ exponentielle Funktionen angenähert. Unter Berücksichtigung von Verwaltungszeit wird eine praktisch optimale Rechnerkernzuteilung entwickelt. Zur Belegung eines Arbeitsspeichers mit zwei Programmplätzen läßt sich eine bezüglich der mittleren Antwortzeit besonders günstige Belegungsstrategie angeben. Die Arbeit zeigt, daß die Simulation zur Optimierung der Ablaufsteuerung in Betriebssystemen geeignet ist.

There are two main problems in operating systems: managing CPU and main memory. We use simulation and computation to develop realistic computable models of multiaccess time sharing systems. Work on tasks is described as a series of transport- and compute times for "subtasks". Very important is the cumulative d. f of subtask compute-times. They are measured and approximated by hyper-, piecewise, and degenerate negativ exponential functions. Considering overhead we develop a nearly optimal CPU scheduling strategy. Main memory with space for exactly two programs is managed to reach an especially favourable mean response time for tasks. It is shown that simulation is suitable to optimize strategies in operating systems.

1. Einleitung

Es ist ein großer Vorteil, wenn ein Teilnehmerrechensystem durch ein wirklichkeitsnahes und analytisch berechenbares Modell nachgebildet werden kann. Simulationsergebnisse solcher Modelle haben den großen Nachteil, daß bei der Vielzahl der Parameter die Aussagen undurchsichtig werden. Bis heute liegen nur Rechenergebnisse für sehr einfache Modelle von Teilnehmerrechensystemen vor (z.B. [4], [9]), die als obere oder untere Abschätzung für interessante Betriebsziele, wie beispielsweise die mittlere Antwortzeit der Benutzer, zweifellos ihre Berechtigung haben. Auch für analytische Aussagen über optimale Strategien zum Einsatz der in beschränkter Zahl vorhandenen Betriebsmittel Rechnerkern, Arbeitsspeicher und Transportkanal mußte man sich auf stark vereinfachte Modelle beschränken [I], [3], [6]. Es besteht deshalb weitgehende Unsicherheit über die Aussagekraft der für Teilnehmersysteme berechneten Resultate.

Diese Arbeit beschreibt, wie Simulationsergebnisse zur Entwicklung von analytisch berechenbaren Modellen genutzt werden können. Die angeführten Beispiele sind Stationen auf dem Lösungsweg der Aufgabe, für ein Teilnehmerrechensystem die mittlere Antwortzeit und im Stapelbetrieb den Durchsatz zu optimieren. Bild 1 zeigt eine schematische Darstellung des Systems. Für die durch ihre statistischen Eigenschaften bekannten Benutzerprogramme sind weder praktikable optimale Zuteilungsstrategien des Rechnerkerns an Programme im Arbeitsspeicher, noch optimale Belegungsstrategien für den begrenzt großen Arbeitsspeicher bekannt.

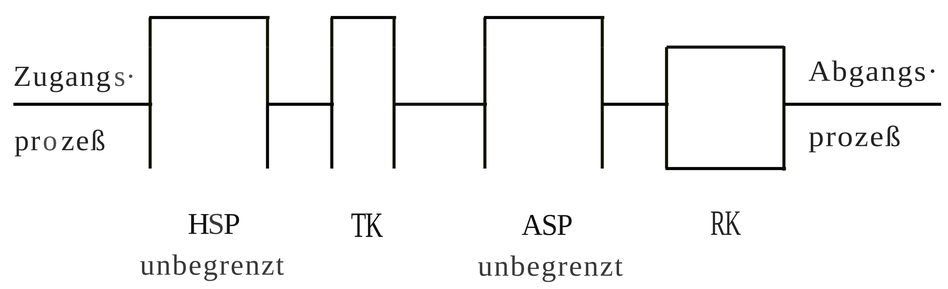

Bild I. Schematische Darstellung des Rechensystems. Gesucht sind optimale Strategien zur Rechnerkernzuteilung an Programme im ASP und optimale Strategien zur Arbeitsspeicherbelegung.
HSP Hintergrundspeicher TK Transportkanal
ASP Arbeitsspeicher RK Rechnerkern

Ausgangspunkt ist das Ergebnis der analytischen Berechnung von optimalen Rechnerkern-Zuteilungsstrategien für ein sehr einfaches Modell [6]. Bei der Suche nach einem möglichst realistischen berechenbaren Modell mit optimalen Rechnerkern- und Arbeitsspeicher-Zuteilungsstrategien hat sich ein zyklischer Weg mit schrittweiser Verfeinerung der Modelle bewährt:

* Die Arbeit ist zu 50% aus Mitteln des Bundesministeriums für Forschung und Technologie (2. DV-Programm, Teil 2.3) gefördert worden.

a) Simulation

Erweiterung des Modells durch Berücksichtigung neuer Parameter (z. B. Größe des Arbeitsspeichers, Typ eines Hintergrundtransportes) und Simulation mit systematischer Untersuchung der Abhängigkeit der mittleren Antwortzeit von allen Parameterkombinationen.

b) Berechnung

Konstruktion eines berechenbaren Modells auf Grund der Erkenntnisse aus a), so daß die Ergebnisse praktisch erhalten bleiben.

c) Modellverfeinerung

Wiederholung von a), b) und c), um ein wirklichkeitsnäheres berechenbares Modell zu erhalten.

2. Programmbearbeitung durch Transporte und Rechnung an Teilaufgaben

Die Bearbeitung eines Programms in einem Rechner zerfällt in typische Unterabschnitte. Es wechseln Rechen- und Transportphasen ab, so daß der Rechnerkern und ein Transportkanal immer abwechselnd arbeiten, solange nur ein Programm zur Bearbeitung ansteht (Bild 2a). Im folgenden wird die Bearbeitung eines Gesamtprogrammes als Aufgabe bezeichnet. (In einem Teilnehmersystem ist eine Aufgabe durch zwei Konsolkontakte des Rechners eingegrenzt.) Die durch den Rechnerkern ununterbrochen fortsetzbare Bearbeitung für eine Aufgabe ist eine Teilaufgabe dieser Aufgabe. Eine Teilaufgabe ist beendet, wenn ein Transport vom/zum Hintergrundspeicher nötig ist, um Befehle oder Daten zur Fortsetzung der Rechenarbeit an dieser Aufgabe zu transportieren. Erst nach Beendigung des Transportes kann die nächste Teilaufgabe gerechnet werden. Nach Bild 2a ist für das Laden eines Programms und damit der anschließend bearbeitbaren

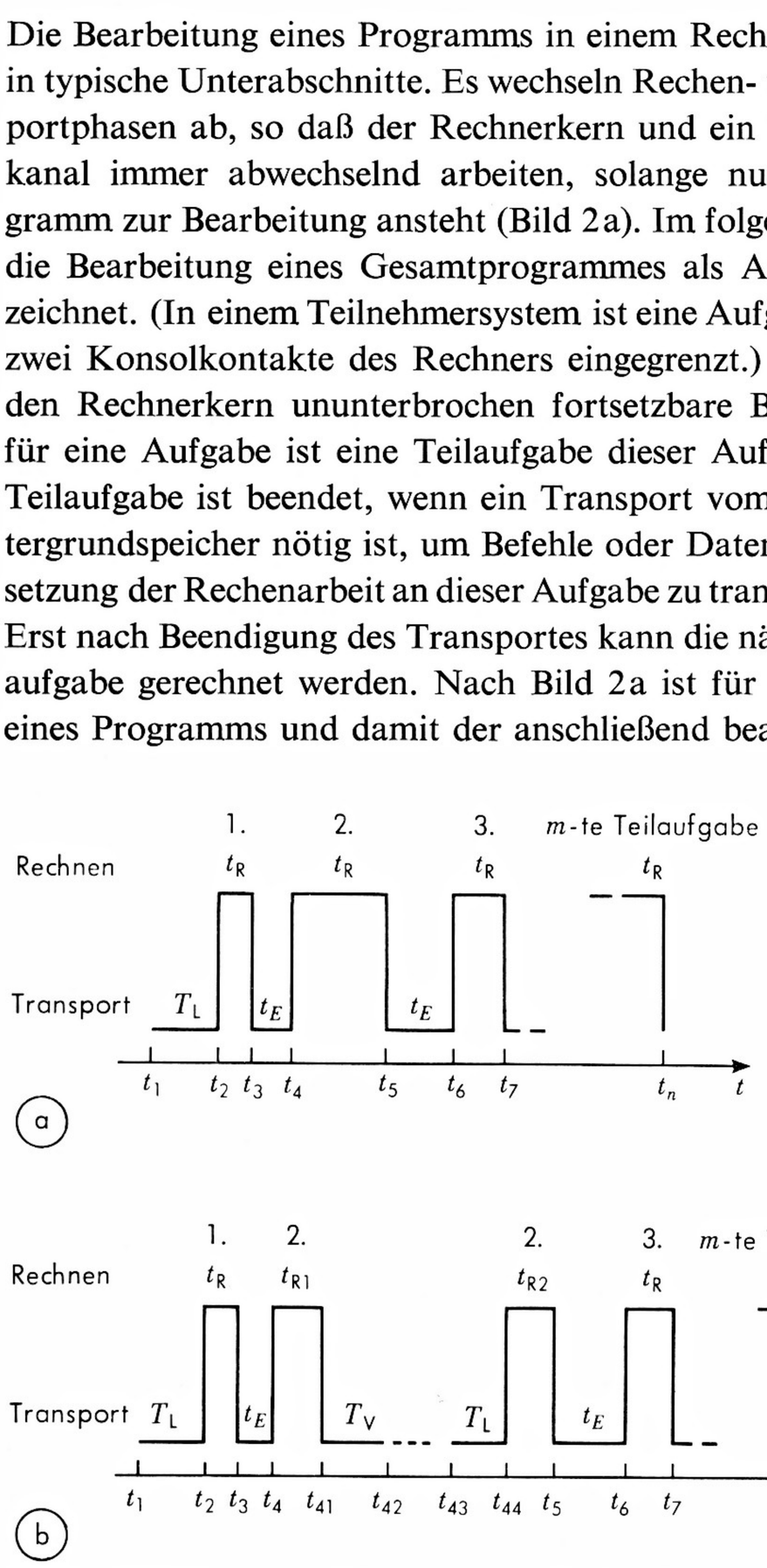

Bild 2. Definition von Transport- und Rechenzeiten einer Aufgabe.
a) Ungestörte Bearbeitung einer Aufgabe
b) Einmal aus dem Arbeitsspeicher verdrängte Aufgabe.
T_L Ladetransport einer Aufgabe, T_V Verdrängungstransport einer Aufgabe, t_E Ergänzungstransport der nächsten Teilaufgabe, t_R Rechenzeit einer Teilaufgabe.

Teilaufgabe die Zeit T_L nötig, während für Ergänzungstransporte zur Aufbereitung der nächsten Teilaufgabe einer Aufgabe die Zeit t_E erforderlich ist.

Eine Teilaufgabe ist nicht beendet, wenn die Rechenarbeit an ihr unterbrochen wird, um für eine Teilaufgabe einer anderen Aufgabe zu arbeiten. Die bereits angefangene Teilaufgabe wird später weitergerechnet. Während einer solchen Unterbrechung kann es vorkommen, daß die Befehle und Daten der betroffenen Aufgabe in den Hintergrundspeicher verdrängt werden. Nach Bild 2b beansprucht jede Verdrängung eines Programms die Zeit T_V. Das Wiederladen des Programms mit seiner aktuellen Teilaufgabe verbraucht die Zeit T_L.

Messungen in einem Rechner haben ergeben, daß die Erwartungswerte der Variablen T_L und T_V ungefähr gleich groß sind $E(T_L) \approx E(T_V)$, und daß Ergänzungstransporte im Mittel weniger Zeit als Lade- bzw. Verdrängungstransporte beanspruchen $E(t_E) < E(T_L)$. Es hat sich außerdem gezeigt, daß die Transportzeiten T_L, T_V, t_E als Zufallsvariable aus einer negativ exponentiellen Verteilung

$$P(t_H \leq t) = 1 - e^{-t/E(t_H)} \tag{2.1}$$

aufgefaßt werden können. Abhängigkeiten zwischen den Rechenzeiten einer Teilaufgabe t_R und Transportzeiten t_H sind praktisch nicht feststellbar. Die einzelnen Verteilungen ergeben sich durch Ersetzen von t_H durch t_E bzw. T_L bzw. T_V.

Die Zerstückelung der Aufgabe in Teilaufgaben wird u. a. wesentlich durch das Betriebssystem verursacht. Für Stapelverarbeitung wurden im Teilnehmerrechensystem TNS 440 der *Telefunken-Computer* ähnlich wie in einem /360-System [5] 100 und mehr Teilaufgaben pro Aufgabe gemessen [10]. Für den Teilnehmerbetrieb liegen noch keine Messungen vor, man muß dort jedoch mit wesentlich weniger Teilaufgaben pro Aufgabe rechnen (z. B. 10).

Es ist angenommen, daß die Ein-/Ausgabe von Programmen zwischen den Peripheriegeräten und dem Hintergrundspeicher (Trommel, Platte) gesondert betrachtet werden kann. Alle Überlegungen zur Optimierung der mittleren Antwortzeit erstreckten sich in dieser Arbeit nur über die Zeit vom Vorhandensein der bearbeitbaren Aufgabe im Hintergrundspeicher bis zu ihrer Fertigstellung durch den Rechnerkern und den Transportkanal, so daß die Aufgabe wieder, aber fertiggestellt, im Hintergrundspeicher liegt.

2.1 Rechenzeitverteilung der Teilaufgaben

Alle bekannten theoretischen Arbeiten über optimale Zuteilung des Rechnerkerns und zur Berechnung von Optimierungszielen (z. B. mittlere Antwortzeit) bei gegebener Rechnerkernzuteilung setzen voraus, daß die durch eine Wahrscheinlichkeitsverteilung gegebene Rechenzeit zusammenhängend abgearbeitet werden kann. Dazwischengeschobene Transporte sind nicht zugelassen. Es ist offensichtlich, daß diese Arbeiten für Teilaufgaben aber nicht für Aufgaben anwendbar sind.

Messungen der Rechenzeiten t_R von Teilaufgaben sind aus der Literatur nicht bekannt. Deshalb wurde die Summen-

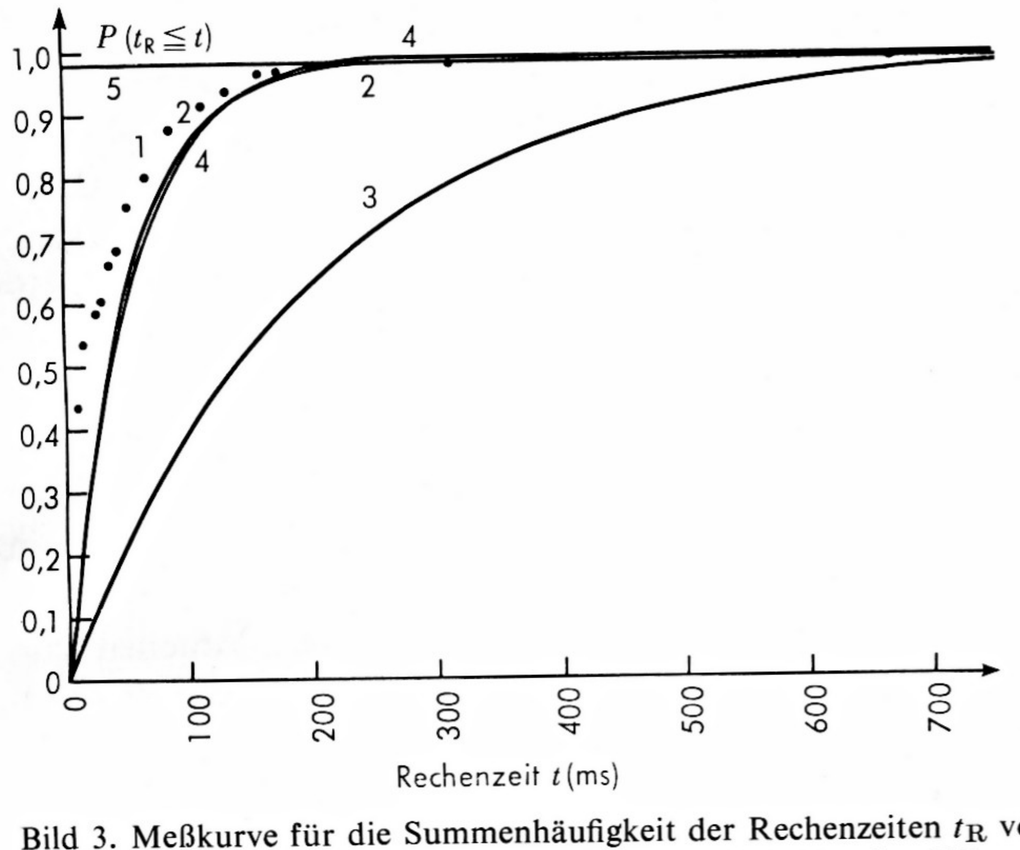

Bild 3. Meßkurve für die Summenhäufigkeit der Rechenzeiten t_R von Teilaufgaben im Stapelbetrieb des Systems TNS 440 der TC und Approximation durch verschiedene Verteilungsfunktionen gleichen Erwartungswertes und (bei Kurven 2, 4 und 5) auch mit gleichem Variationskoeffizienten.

1 Meßkurve, Mittelwert 0,2 s, Streuung 4,5 s^2;
2 Näherung: Hyperexponentielle Verteilung Gl. (2.4) $\mu_1 = 19{,}73\ s^{-1}$, $\mu_2 = 0{,}066\ s^{-1}$, $w_1 = 0{,}99$;
3 Näherung: Exponentialverteilung Gl. (2.3) $\mu = 5\ s^{-1}$;
4 Näherung: Stückweise exponentielle Verteilung Gl. (4.2) $\mu_1 = 18{,}4\ s^{-1}$, $\mu_2 = 0{,}066\ s^{-1}$, $t_g = 0{,}25$ s, $P_g = 0{,}99$;
5 Näherung: Entartete Exponentialverteilung Gl. (4. 11) $P_g = 0{,}983$, $\mu_2 = 0{,}085\ s^{-1}$.

häufigkeit der Rechenzeiten von Teilaufgaben an der TC-TR 440 gemessen. Es ergeben sich sowohl für Teilnehmer- als auch für Stapelbetrieb ähnliche Kurvenverläufe, denen gemeinsam ist, daß ein hoher Prozentsatz der Teilaufgaben sehr wenig Rechenzeit und der Rest vergleichsweise sehr viel beanspruchen. Der Variationskoeffizient

$$VK = \sigma(t_R)/E(t_R), \tag{2.2}$$

($\sigma^2(t_R)$ = Varianz; $E(t_R)$ = Rechenzeiterwartungswert einer Teilaufgabe), ist deutlich größer als bei einer negativ exponentiellen Verteilung und für Teilaufgaben im Stapelbetrieb besonders groß[1]). Nach Bild 3 wird deutlich, daß die häufig als Näherung vorgeschlagene Exponentialverteilung

$$P(t_R \leq t) = 1 - e^{-t/E(t_R)} \tag{2.3}$$

unzureichend ist. Der gemessene Kurvenverlauf kann dagegen gut durch eine Hyperexponentialverteilung

$$P(t_R \leq t) = 1 - \sum_{i=1}^{n} w_i e^{-\mu_i t}, \quad \mu_i > 0;\ w_i > 0;\ \sum_{i=1}^{n} w_i = 1; \tag{2.4}$$

approximiert werden. Ihr Erwartungswert und ihre Varianz berechnen sich aus

$$E(t_R) = \sum_{i=1}^{n} w_i/\mu_i, \tag{2.5}$$

$$\sigma^2(t_R) = \sum_{i=1}^{n} 2\, w_i/\mu_i^2 - (E(t_R))^2. \tag{2.6}$$

[1]) (z. B. 10)

Der Variationskoeffizient von Kurve 2 in Bild 3 stimmt mit dem der Meßkurve überein. Während die Approximation durch eine Exponentialfunktion nur das 1. Moment der Meßkurve berücksichtigen kann, werden durch Verteilungen nach Gl. (2.3) auch Momente höherer Ordnung erfaßt. In Bild 3 sind noch weitere Näherungen eingetragen, die später besprochen werden.

3. Einfaches Rechnermodell mit bekannter optimaler Rechnerkernzuteilung

Für das sehr einfache Ausgangs-Modell (Bild 4) sind folgende Annahmen gemacht:

- Eine Aufgabe habe genau eine Teilaufgabe;
- Der Rechnerkern kann ohne Zeitverlust von der Bearbeitung einer Aufgabe zu einer anderen übergehen;
- Der Zugangsprozeß liefert, wenn der Arbeitsspeicher leer ist, gleichzeitig eine bekannte Zahl M von Aufgaben;
- Fertiggerechnete Aufgaben verlassen sofort das Modell.

Die Gesamtverweilzeit aller Aufgaben im Modell soll durch Optimierung möglichst klein gemacht werden. Durch die Warteschlangen im Arbeitsspeicher ist angedeutet, daß die vorhandenen Aufgaben klassifiziert werden können. In [6] ist für beliebige Rechenzeitverteilungsfunktionen die jeweils optimale Rechnerkern-Zuteilungsstrategie für dieses Modell angegeben. Danach ist für eine *negativ exponentielle Rechenzeitverteilung* die Reihenfolge der Bearbeitung beliebig und ohne Einfluß auf das Optimierungsziel.

Für eine hyperexponentielle Verteilung muß eingehalten werden, daß alle wartenden Aufgaben bei fortschreitender Bearbeitung die gleiche verbrauchte Rechenzeit ausweisen. Man kommt zu einer *Strategie SO*, die häufig *infinitesimal kleine* Zeitscheiben an konkurrierende Teilaufgaben zuteilen muß. In wirklichen Rechnern ist der Übergang der Bearbeitung von einer zur anderen Teilaufgabe mit Verwaltungsaufwand (overhead) verbunden. Deshalb wird man die Bearbeitung von Teilaufgaben mit exponentiell verteilter Rechenzeit ohne Unterbrechungen durchführen. Bei hyperexponentiell verteilter Rechenzeit erhebt sich die Frage, wie denn die optimale Zeitzuteilung unter Berücksichtigung von Verwaltungszeit τ beim Wechsel der Bearbeitung aussehen muß. Diese Frage kann durch analytische Rechnung nicht beantwortet werden. Deshalb wird durch Simulation geklärt, wie stark sich gegebene Zeitzuteilungsstrategien bezüglich des Optimierungszieles mittlere Antwortzeit unterscheiden, wenn Verwaltungszeit berücksichtigt wird. Neben der für $\tau = 0{,}0$ optimalen Strategie SO werden verschiedene Reihumverfahren, die jeder wartenden Teilaufgabe eine begrenzte Zeitscheibe zuweisen, untersucht. Dabei werden hyperexponentielle Rechenzeitverteilungen mit verschiedenem Variationskoeffizienten betrachtet.

4. Entwicklung praktisch optimaler Rechnerkern-Zuteilungsstrategien bei Verwaltungszeit nach Einführung der stückweise bzw. entartet negativ exponentiellen Verteilungen

4.1 Simulationsmodell zur Beurteilung von Rechnerkern-Zuteilungsstrategien bei Verwaltungszeit

Wir erweitern das Modell nach Bild 4 so, daß Aufgaben (mit je einer Teilaufgabe) nicht mehr nur zu genau festgelegten

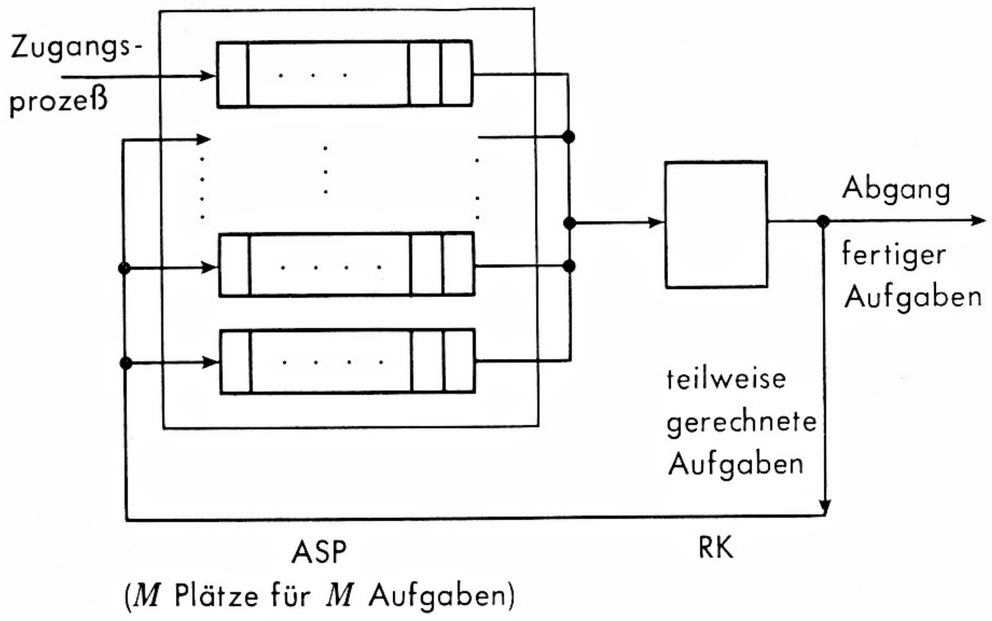

Bild 4. Einfaches Rechnermodell zur analytischen Berechnung der optimalen Rechnerkernzuteilungsstrategie (Stoßzugangsprozeß mit genau M Aufgaben) und zur Simulation von Zuteilungsstrategien (Poisson-Zugang, unbegrenzter Arbeitsspeicher).

Zeitpunkten gebündelt eintreffen, sondern daß Teilaufgaben nach einem Poissonprozeß mit der Zugangsrate λ eintreffen. Der Arbeitsspeicher soll unendlich groß sein, so daß die beiden in Bild 1 gezeigten Bedienstellen Transportkanal und Rechnerkern vollständig entkoppelt sind. Man kann dann die Optimierung der Rechnerkern- und Transportkanal-Zuteilung getrennt betrachten. Aus der Zugangsrate λ und dem Rechenzeiterwartungswert $E(t_R)$ von Teilaufgaben bestimmt man die Wahrscheinlichkeit A (auch Angebot genannt), daß der Rechnerkern Arbeit hat zu

$$A = \lambda \cdot E(t_R). \tag{4.1}$$

Abhängig vom Angebot stellt sich eine mittlere Antwortzeit $E(t_A)$ für Teilaufgaben ein. Die Antwortzeit t_A erstreckt sich über die Zeit vom Eintreffen der Teilaufgabe im Arbeitsspeicher bis zur Fertigstellung durch den Rechnerkern. Im Modell (Bild 4) werden die Teilaufgaben nach schon verbrauchter Rechenzeit in verschiedene Warteschlangen geordnet.

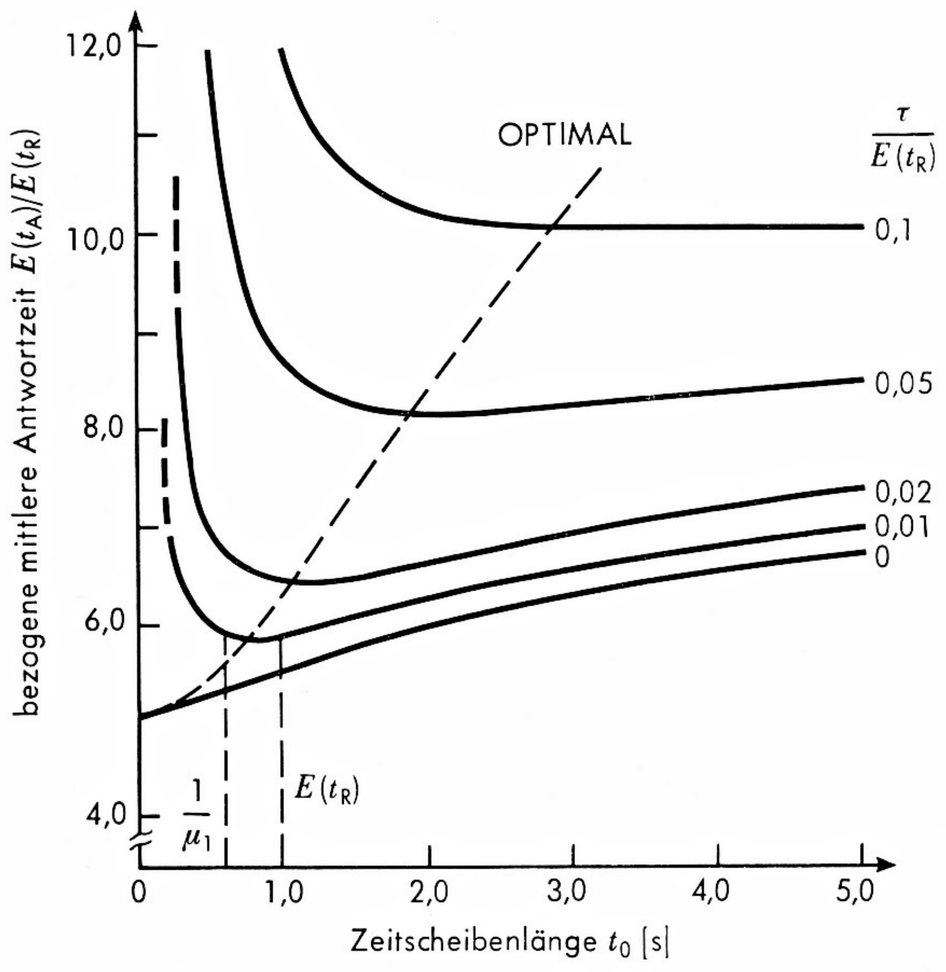

Bild 5. Round Robin Rechnerkernzuteilung mit Verwaltungszeit τ bei hyperexponentieller Rechenzeitverteilung Gl. (2.4). Ergebnis für ein Angebot $A = 0{,}8$ aus [7]. $E(t_R) = 1$ s, $\sigma^2/(t_R) = 3\,s^2$, $w_1 = 0{,}7887$ $\mu_1 = 1{,}5774\ s^{-1}$, $\mu_2 = 0{,}644\ s^{-1}$.

Jedesmal, wenn der Rechnerkern aufgrund der vorgeschriebenen Strategie die Bearbeitung einer Teilaufgabe zugunsten einer anderen abbricht, wird der dabei nötige Verwaltungsaufwand durch eine konstante Zeit τ berücksichtigt. Jeder Verwaltungsakt verzögert die Fertigstellung aller Wartenden. Für *zyklische Rechnerkern-Zuteilungsstrategien* (Round Robin) erhält man bei gegebenem Angebot A abhängig von der Größe der Verwaltungszeit τ eine besonders kleine mittlere Antwortzeit, wenn die Zeitscheibenlänge geeignet gewählt wird. Es ist inzwischen gelungen, die mittlere Antwortzeit als Funktion des Angebotes zu berechnen [7]. Bild 5 zeigt ein entsprechendes Ergebnis. Die Parameter der untersuchten Hyperexponentialfunktion (Gl. 2.4) mit zwei Anteilen sind in Bild 5 angegeben. Schon bei sehr kleinen normierten Verwaltungszeiten $\tau/E(t_R)$ muß eine Rechenzeitscheibe in der Größenordnung des Rechenzeiterwartungswertes $E(t_R)$ angewandt werden.

Bei der Interpretation des Ergebnisses in Bild 5 stößt man auf eine wichtige Forderung an günstige Zeitzuteilungs-Strategien bei Verwaltungszeit: Die Zeitscheibe ist so zu wählen, daß während jeder Scheibe ein spürbarer Prozentsatz von Teilaufgaben fertiggestellt wird. Aus Bild 5 ersieht man für sehr kleine Verwaltungszeit ($\tau/E(t_R) = 0{,}01$), daß die optimale Zeitscheibe bereits größer sein muß, als der Rechenzeiterwartungswert $1/\mu_1$ des Anteils 1 der hyperexponentiellen Verteilung. Nach dieser ersten Zeitscheibe liegt eine Restrechenzeitverteilung mit wesentlich kleinerer Streuung als vorher vor. Nach weiteren Zeitscheiben bleibt praktisch eine Exponentialfunktion mit dem Erwartungswert $1/\mu_2$ übrig. Diese Restverteilung darf nach Abschnitt 3 nur dann mit Zeitscheiben abgearbeitet werden, wenn $\tau = 0{,}0$ ist.

Wir haben deshalb *Strategien SN* untersucht, die eine begrenzte Zahl N von Zeitscheiben gleicher Länge t_0 pro Teilaufgabe zulassen, und danach nicht fertiggestellte Teilaufgaben (Langrechner) mit unbegrenzter Rechenfrist dann bearbeiten, wenn nur Langrechner warten. Dann wird jeweils der Langrechner mit kleinster verbrauchter Rechenzeit ausgewählt. Neu eintreffende Teilaufgaben erhalten unterbrechende Priorität gegenüber Langrechnern.

Wie sich zeigt, führt schon bei kleiner normierter Verwaltungszeit die Zuteilung *genau einer* Zeitscheibe der Dauer $t_0 = t_g$ zur minimalen mittleren Antwortzeit, so daß dann die *Strategie S1* optimal ist. Je größer der Variationskoeffizient der Rechenzeitverteilung ist, um so kleiner darf $\tau/E(t_R)$ sein, ohne daß die Strategie S1 von einer anderen übertroffen wird. Die mittlere Antwortzeit wird unter der Strategie S1 deutlich kleiner, als mit der Strategie RR bei günstigster Zeitscheibe t_0 (vgl. Bild 5). Bild 6 zeigt ein Simulationsergebnis für die in Bild 3 angegebene hyperexponentielle Verteilung. Danach ist eine sehr große Zeitscheibe, bei der die Strategie S1 in eine Bearbeitung „nach der Reihenfolge des Eintreffens" übergeht, ähnlich wie in Bild 5, ungünstig. Bei sehr kleiner Zeitscheibenlänge z. B. $\{t_g = 0{,}1\ E(t_R)\}$ wird die bezogene mittlere Antwortzeit gegenüber dem bei optimaler Frist erreichten Minimum bei weitem nicht so stark vergrößert, wie bei der Strategie RR (Bild 5). Das liegt daran, daß dort wesentlich mehr Verwaltungsakte nötig sind. Wegen der unterbrechen-

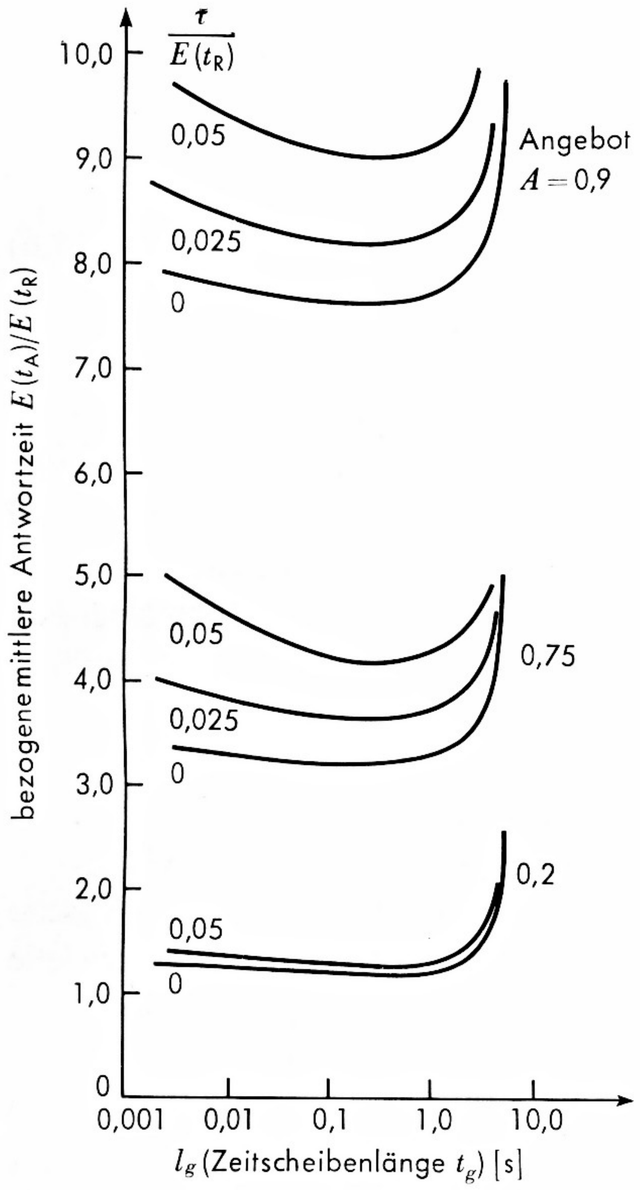

Bild 6. Simulationsergebnisse bei hyperexponentieller Rechenzeitverteilung (Bild 3) unter der Strategie S1 bei verschiedenen Angeboten A. Berücksichtigt man eine kleine Verwaltungszeit τ beim Wechsel der Bearbeitung, so ergibt sich ein Minimum $0{,}2 \leq t_g \leq 0{,}5$ s.

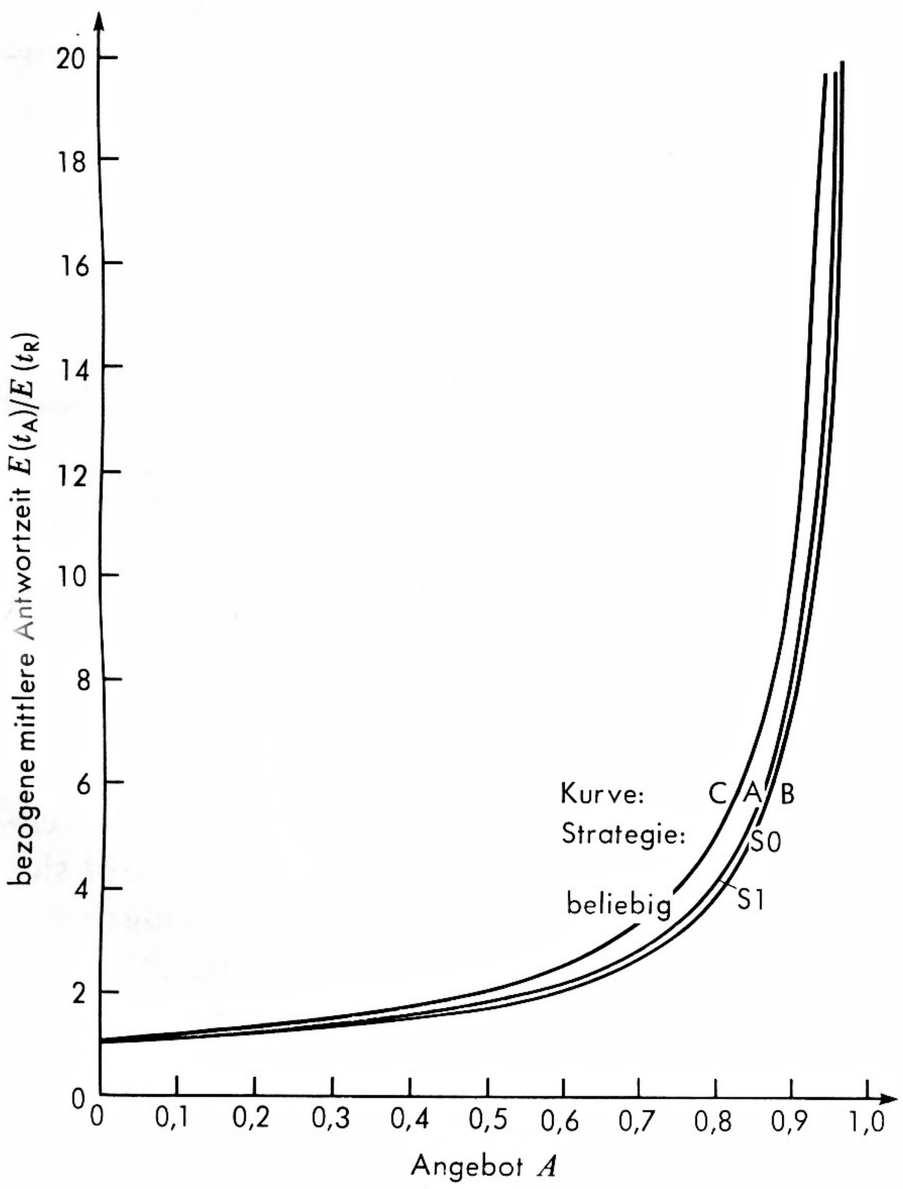

Bild 7. Modell nach Bild 4 mit unbegrenztem ASP. Hyperexponentielle Rechenzeitverteilung für Kurve A und B (Bild 3), und negativ exponentielle Verteilung für Kurve C. Verwaltungszeit $\tau = 0{,}0$. Vergleich zweier Strategien SO (unendlich oft unterbrechen) und S1 (höchstens eine Unterbrechung pro Teilaufgabe).

den Priorität neu im Arbeitsspeicher eingetroffener Teilaufgaben wird durch die Strategie S 1 genügend gut erreicht, daß Teilaufgaben mit kleiner verbrauchter Rechenzeit vor solchen mit größerer verbrauchter Rechenzeit bearbeitet werden.

Interessant ist nun noch der Vergleich der Strategien SO (optimal bei $\tau = 0{,}0$) und S 1 bezüglich der bei $\tau = 0{,}0$ erzielten mittleren Antwortzeit. Bild 7 zeigt, daß bei $\tau = 0{,}0$ die Strategie S 1 nur unwesentlich schlechter als die Strategie SO ist, wenn die Hyperexponentialfunktion nach Bild 3 angesetzt wird. In Betriebssystemen von Rechenanlagen treten beim Wechsel der Bearbeitung Verwaltungszeiten auf. Man wird dort deshalb anstelle der für $\tau = 0{,}0$ optimalen Strategie SO die durch Simulation gefundene praktisch optimale Strategie S 1 verwenden. Die Strategie S 1 ist anzuwenden, solange nicht wegen zu großer Verwaltungszeit τ eine Abarbeitung ganz ohne Unterbrechungen noch günstiger wird.

Für einen bekannten Benutzerkreis einer Rechenanlage ändert sich laut Messung in Rechenanlagen der Variationskoeffizient und der Erwartungswert $E(t_R)$ der Teilaufgaben-Rechenzeitverteilung abhängig von der Tageszeit. Die Verwaltungszeit τ bleibt konstant. Je größer der Variationskoeffizient ist, um so kleiner darf $\tau/E(t_R)$ sein, ohne daß eine Strategie SN $(N > 1)$ zu einer kleineren mittleren Antwortzeit als die Strategie S 1 führt. Wählt man als Näherungsfunktion für die Rechenzeitverteilung eine hyperexponentielle Verteilung, so führt das generell zu Unklarheiten darüber, wann abhängig von $\{ \tau/E(t_R), \text{VK} \}$ welche Strategie SN $(N = 1, 2...)$ und mit welcher Zeitscheibenlänge t_0 zu verwenden ist. Die Hyperexponentialfunktion als Näherung an gemessene Teilaufgaben-Rechenzeitstatistiken führt dazu, daß die Angabe einer optimalen Rechnerkernzuteilungs-Strategie bei Verwaltungszeit sehr erschwert wird.

4.2 Verbessertes analytisches Modell durch Einführung der stückweise negativ exponentiellen Verteilung

Die vorangegangene Diskussion der Simulationsergebnisse hat erbracht, daß es darauf ankommt, Teilaufgaben mit kurzen Rechenzeiten schnell zu entdecken und bevorzugt zu bedienen. Dafür reicht eine grobe Unterteilung aller Teilaufgaben in zwei Klassen aus. Wir suchen deshalb eine Näherung an die statistische Verteilung der Teilaufgabenrechenzeiten (Bild 3), die eine Klassenbildung erleichtert. Für die Konstruktion der neuen Funktion wird die Kenntnis genutzt, daß bei negativ exponentiell verteilter Rechenzeit alle denkbaren Strategien die gleiche mittlere Antwortzeit erzielen. Bei hyperexponentieller Rechenzeitverteilung und $\tau/E(t_R) > 0$ hatten bei Anwendung der dann optimalen Strategie S 1 mit der Zeitscheibenlänge t_g Langrechnerteilaufgaben $(t_R > t_g)$ näherungsweise eine negativ exponentielle Restrechenzeitverteilung. Ihre Rechenzeit wird jetzt exakt durch eine Exponentialfunktion beschrieben. Die neue Verteilung soll die Eigenschaft haben, daß die Strategie S 1 optimal ist, solange unterbrechende Strategien überhaupt günstiger als nicht unterbrechende abschneiden. Teilaufgaben mit verbrauchter Rechenzeit $t_R < t_g$ müssen dann ebenfalls bezüglich der Bearbeitungsreihenfolge gleichwertig sein. Zwischenergebnisse aus [6] führen unter Berücksichtigung der für die ge-

suchte Funktion genannten Bedingungen zu der erstmals von *Marte* [2] veröffentlichten neuen Verteilung. Die Funktion ist in [9] allgemein für $(j+1)$ exponentielle Anteile $(j=0, 1, ...)$ beschrieben. Dort ist auch gezeigt, daß es für die hier betrachteten Rechenzeitverteilungen von Teilaufgaben ausreicht, wenn man als Näherung die stückweise exponentielle Verteilung mit nur zwei Anteilen auswählt

$$P(t_R \leq t) = \begin{cases} 1 - e^{-\mu_1 t} & | 0 \leq t \leq t_g, \\ 1 - e^{-(\mu_1 - \mu_2) t_g - \mu_2 t} & | t_g < t. \end{cases} \qquad (4.2)$$

$$\mu_1 > \mu_2 > 0$$

Nach [9] ist die mittlere Antwortzeit von Teilaufgaben mit einer nach Gl. (4.2) beschriebenen Rechenzeit unter der Strategie S1 relativ unempfindlich gegen eine zu groß gewählte Zeitscheibe t_g. Man darf demnach in weiten Grenzen unabhängig von Schwankungen der Rechenzeitverteilung die Strategie S1 mit einer genügend groß gewählten Zeitscheibe t_g verwenden.

In Bild 3 ist eine solche Funktion eingetragen worden. Im Rahmen der geforderten Genauigkeit der Approximation kann man sagen, daß die Meßkurve genügend gut angenähert wird. Hyperexponentielle und stückweise exponentielle Verteilungsfunktionen sind gleichgut als Näherungen geeignet. Die folgenden Beispiele zeigen, daß die Einführung der neuen Funktion einen deutlichen Fortschritt auf dem Gebiet der Analyse von Modellen, wie der Synthese von Strategien bei gegebenem Optimierungsziel ermöglicht.

Für das bisher betrachtete Rechnermodell mit unbegrenztem Arbeitsspeicher kann die mittlere Antwortzeit für Teilaufgaben mit stückweise exponentieller Rechenzeitverteilung bei Verwaltungszeit $\tau = 0{,}0$ und optimaler Strategie S1 nach [9] berechnet werden. Man erhält für beide Strategien SO und S1 dasselbe Resultat. Es deckt sich weitgehend mit dem Simulationsergebnis bei hyperexponentieller Rechenzeitverteilung unter der Strategie S1 (Kurve A, Bild 7).

4.3 Modifikation des analytischen Modells durch Einführung der entartet negativ exponentiellen Verteilung

Bei der Betrachtung der Ergebnisse für unterschiedliche stückweise exponentielle Verteilungen, die sich als Näherungen für verschiedene Meßkurven der Summenhäufigkeiten von Teilaufgabenrechenzeiten ergaben, fällt eine typische Eigenschaft auf. Bei jedem Angebot unterscheiden sich die mittleren Antwortzeiten für eine stückweise exponentielle Rechenzeitverteilung bei optimaler Strategie S1 und für eine Exponentialrechenzeitverteilung bei beliebiger Strategie nur wenig (vgl. Beispiel in Bild 7, Kurve A für stückweise expon. Verteilung nach Bild 3 und Kurve C).

Der Rechenzeit-Erwartungswert $E(t_R)$ bei stückweise exponentieller Verteilung

$$E(t_R) = 1/\mu_1 + e^{-\mu_1 t_g}(1/\mu_2 - 1/\mu_1) \qquad (4.3)$$

setzt sich aus dem Erwartungswert der Teilaufgaben mit Rechenzeiten $0 \leq t \leq t_g$

$$E(t_R)_{kurz} = \frac{1}{\mu_1} - \frac{t_g}{e^{\mu_1 t_g} - 1} \qquad (4.4)$$

und dem Erwartungswert der Langrechner mit $t_g < t$

$$E(t_R)_{lang} = t_g + 1/\mu_2 \qquad (4.5)$$

zusammen.

$$E(t_R) = P_g E(t_R)_{kurz} + (1 - P_g) E(t_R)_{lang}. \qquad (4.6)$$

P_g ist die Wahrscheinlichkeit dafür, daß eine unbearbeitete Teilaufgabe kurz ist

$$P_g = 1 - e^{-\mu_1 t g} \qquad (4.7)$$

Setzt man Zahlenwerte für das Beispiel aus Bild 3 ein, so wird deutlich, daß die Langrechner trotz ihrer geringen Häufigkeit von nur 1% aller Teilaufgaben mehr als 75% der gesamten Rechenzeit verbrauchen. Für andere hier nicht aufgeführte Messungen der Rechenzeitverteilungen (besonders im Stapelbetrieb) ergeben sich je nach Aufgabenprofil und Rechenzentrum zum Teil prozentual noch höhere Anteile der Langrechner-Rechenzeiten an der gesamten Rechenzeit $E(t_R)$. Diese Erkenntnis läßt sich ausnutzen, um die für analytische Rechnungen relativ unhandliche stückweise exponentielle Verteilung zu vereinfachen.

Die stückweise exponentielle Verteilung ist so konstruiert worden, daß Langrechner-Teilaufgaben (verbrauchte Rechenzeit $t_R \geq t_g$) eine exponentielle Rechenzeitverteilung mit dem Erwartungswert

$$E(t_R \mid t > t_g) = 1/\mu_2 \qquad (4.8)$$

haben (vgl. Gl. (4.5)). Für nur solche Teilaufgaben würde sich Kurve C in Bild 7 ergeben. Die bevorzugte Bearbeitung unbearbeiteter Teilaufgaben durch die Strategie S1 bewirkt eine Verkleinerung der mittleren Antwortzeit gegenüber dem Ergebnis bei exponentieller Rechenzeitverteilung. Nach [9] ergibt sich bei zwei Anteilen der stückweise exponentiellen Verteilung die mittlere Antwortzeit zu

$$E(t_A) = \frac{E(t_R)}{1-A} - \frac{E(t_R)_{kurz}}{1-A}\left\{1 - \frac{1 - P_g \lambda E(t_R)}{1 - P_g \lambda/\mu_1}\right\}. \qquad (4.9)$$

Für eine Exponentialverteilung ist

$$E(t_A) = \frac{E(t_R)}{1-A}. \qquad (4.10)$$

Der Subtrahent in Gl. (4.9) ist immer positiv.

Ist der Beitrag der Kurzrechner zur gesamten mittleren Rechenzeit $E(t_R)$ klein, so darf man die Rechenzeiten kurzer Teilaufgaben $(0 < t_R \leq t_g)$ näherungsweise vernachlässigen $(t_g = 0{,}0)$. Der prozentuale Anteil an der Gesamtzahl der Teilaufgaben bleibt dabei ungefähr erhalten. P_g verschiebt sich leicht, wenn die 1. und 2. Momente der Meßkurve eingehalten werden sollen. Bei dieser Vereinfachung geht die stückweise exponentielle Rechenzeit-Verteilung in die entartete Exponentialverteilung [3] über.

$$P(t_R \leq t) = 1 - (1 - P_g) e^{-\mu_2 t}. \qquad (4.11)$$

Der Erwartungswert bzw. die Streuung berechnen sich aus

$$E(t_R) = (1 - P_g)/\mu_2 \qquad (4.12)$$

$$\sigma^2(t_R) = (1 - P_g^2)/\mu_2^2. \qquad (4.13)$$

Eine Funktion nach Gl. (4.11) ist in Bild 3 eingezeichnet. Selbstverständlich bleibt die Zeitzuteilungsstrategie S1 auch jetzt optimal. Die mittlere Antwortzeit $E(t_A)$ kann als Funktion des Angebotes von Aufgaben (mit je einer Teilaufgabe) berechnet werden. Es ergibt sich unabhängig vom Prozentsatz $P_g \cdot 100\%$ der kurzen Teilaufgaben dasselbe Resultat, wie für eine exponentielle Rechenzeitverteilung (Gl. (4.10))[2]). Die entartete Exponentialverteilung beschreibt den Rechenzeitbedarf von Teilaufgaben mit der Wahrscheinlichkeit P_g durch die Zeit $t_R = 0{,}0$, sonst durch eine Zufallsvariable einer Exponentialverteilung. Die Verteilung ist für überschlägige Berechnungen bei komplizierten Modellen geeignet, denn sie hat viele Eigenschaften der Exponentialfunktion [12].

Damit gelingt es, Simulationsergebnisse bei experimentell gefundenen günstigsten Strategien so zu interpretieren, daß ein berechenbares Modell, z. B. [9], mit bekannter optimaler Strategie entsteht.

5. Entwicklung einer für die mittlere Antwortzeit besonders günstigen Arbeitsspeicher-Belegungsstrategie unter vereinfachten Modell-Annahmen

5.1 Simulationsmodell zur Untersuchung von Arbeitsspeicher-Zuteilungsstrategien

Im nächsten Schritt soll berücksichtigt werden, daß der Arbeitsspeicher begrenzt groß ist. Es wird vereinfachend angenommen, daß alle Programme gleichgroß sind. Transporte aus dem Hintergrund- in den Arbeitsspeicher können nur erfolgen, wenn mindestens ein Platz frei ist. Von den beiden Hauptaufgaben der Betriebsmittelverwaltung kann die zweite nur untersucht werden, wenn man Transportzeiten berücksichtigt. Es ist plausibel, daß die für unbegrenzten Arbeitsspeicher zur Bearbeitung von Teilaufgaben mit stückweise exponentieller Rechenzeitverteilung optimale Strategie S1 auch bei begrenztem Arbeitsspeicher gilt. Ein Beweis dafür ist bisher nicht gelungen. Bild 8 zeigt ein Simulationsmodell für die Bearbeitung von Aufgaben unter Berücksichtigung der Festlegungen in Bild 2. Es wird ein Poisson-Zugangsprozeß von Aufgaben in den Hintergrundspeicher angenommen. Außerdem ist vorausgesetzt, daß Transport- und Rechenzeiten statistisch unabhängig sind. Zunächst betrachten wir den Fall, daß alle Transportzeiten derselben negativ exponentiellen Verteilung mit dem Erwartungswert $E(t_H) = E(T_L) = E(T_V) = E(t_E)$ entstammen. Aufgaben sollen genau eine Teilaufgabe haben.

[2]) wenn die Strategie S1 angewandt wird.

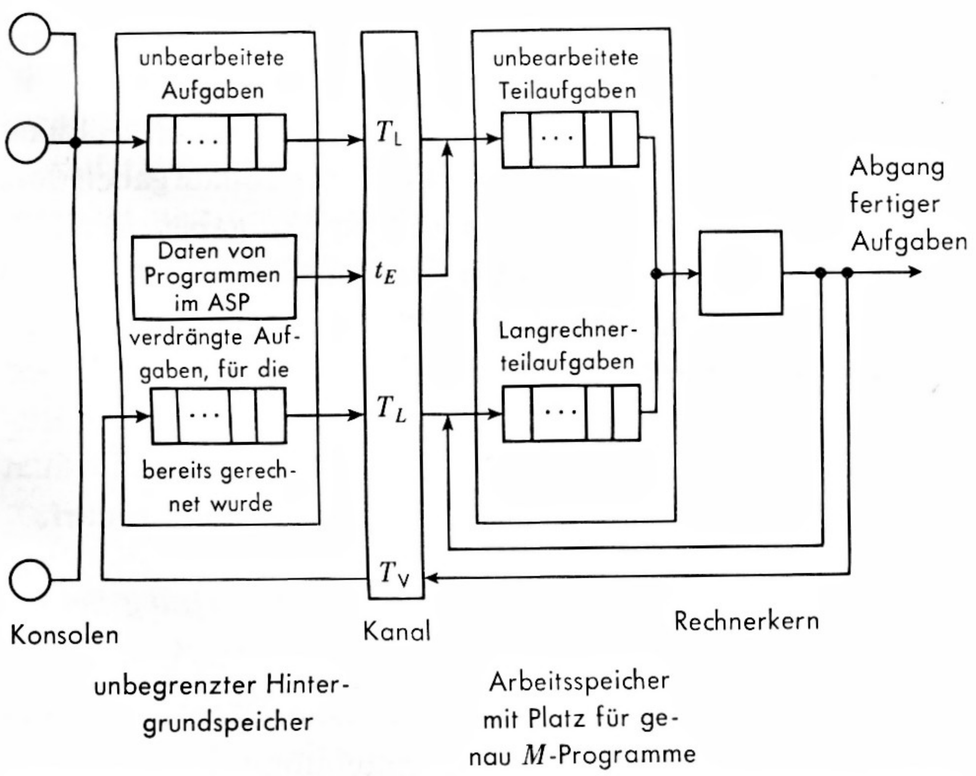

Bild 8. Simulationsmodell des Rechensystems mit begrenztem Arbeitsspeicher. Von den Konsolen treffen Aufgaben ein, die in der Regel mehr als eine Teilaufgabe haben. Nach Bearbeitung einer Teilaufgabe durch den Rechnerkern ist bekannt, ob die Aufgabe eine weitere Teilaufgabe hat und deshalb die Transportzeit t_E benötigt, um die nachfolgende unbearbeitete Teilaufgabe in den Arbeitsspeicher zu laden.

Die Kenntnis der Zeitzuteilung bei begrenztem Arbeitsspeicher macht es möglich, eine günstigste Platzzuteilungsstrategie experimentell durch Vergleich der Resultate verschiedenster Strategien festzulegen. Dabei kommt man abhängig vom Typ der stückweise negativ exponentiellen Verteilung nach [8] mit zwei oder drei Programmplätzen im Arbeitsspeicher aus, wenn die Rechnerkern-Zuteilungsstrategie S1 angewandt wird. Die Arbeitsspeicher-Belegungsstrategie muß dafür sorgen, daß möglichst eine Teilaufgabe, die nicht mit der Frist t_g ausgekommen ist, im Arbeitsspeicher verfügbar ist. Ein solcher Langrechner wird nur dann bearbeitet, wenn keine unbearbeiteten Teilaufgaben im Arbeitsspeicher sind. Zwei solche Langrechner dürfen nur dann gleichzeitig im Arbeitsspeicher bleiben, wenn (im jetzt betrachteten Modell) die mittlere Transportzeit $E(t_H)$ in die Größenordnung des Rechenzeiterwartungswertes der Langrechnerteilaufgaben kommt.

Die mittlere Antwortzeit einer Teilaufgabe erstreckt sich jetzt über die Zeit vom Eintreffen im Hintergrundspeicher bis zur Fertigstellung durch den Rechnerkern im Arbeitsspeicher. Bild 9 zeigt ein Simulationsergebnis für die stückweise exponentielle Rechenzeitverteilung nach Bild 3 bei einem Verhältnis von mittlerer Transport- zu mittlerer Rechenzeit $E(t_H)/E(t_R) = 0{,}5$. Es ergibt sich, daß zwei Plätze im Arbeitsspeicher bei optimaler Zeit- und günstigster Platzzuteilung zu einem ähnlich guten Ergebnis führen wie ein unbegrenzter Arbeitsspeicher.

Mit Hilfe der entarteten Exponentialverteilung kann man ein berechenbares Modell konstruieren, das für einen Sonderfall dieser Verteilung das gefundene Simulationsergebnis bestätigt.

5.2 Analytisches Modell für einen Sonderfall der entartet negativ exponentiellen Rechenzeitverteilung

Die entartete Exponentialverteilung (Gl. (4.7)) wird durch eine extreme Parameterkombination beschrieben: bei konstantem Rechenzeiterwartungswert $E(t_R)$ soll die Wahrscheinlichkeit für kurze Teilaufgaben gegen 1.0 gehen ($P_g \rightarrow 1{,}0$). Dann geht der Rechenzeiterwartungswert der Langrechner $E(t_R/t > t_g)$ (Gl. (4.8)) gegen unendlich. Es soll $E(t_R/t > t_g) \gg E(t_H)$ erfüllt sein. Die günstigste Platzzuteilungsstrategie aus [8] schreibt vor, daß nie zwei Langrechner in den Arbeitsspeicher geladen werden dürfen. Jeder Langrechner wird höchstens einmal verdrängt, nämlich dann, wenn sich eine unbearbeitete Teilaufgabe als Langrechner entpuppt und schon ein Langrechner im Arbeitsspeicher ist. Die Antwortzeit jedes Langrechners wird also höchstens um

$E(t_H) \ll E(t_R/t > t_g)$ vergrößert, also um einen vernachlässigbaren Betrag.

Unbearbeitbare Teilaufgaben werden immer geladen, wenn der zweite Platz im Arbeitsspeicher frei ist und sie werden, dort angekommen, sofort bearbeitet und in der Regel fertig, so daß der Kanal gleich die nächste nachladen kann. Die Tatsache, daß nur ein Platz für unbearbeitete Teilaufgaben zur Verfügung steht, stört nur dann, wenn zwei Langrechner im Arbeitsspeicher sind und erst einer verdrängt werden muß. Wegen $(1\text{-}P_g) \ll 1{,}0$ gibt es nur sehr wenig lange Teilaufgaben, so daß diese Störung sehr selten auftritt. Man sieht, daß zwei Plätze in den weitaus meisten Fällen ausreichen, ohne den Bearbeitungsablauf irgendwie zu stören. Treten gar keine Störungen auf, so ist die mittlere Antwortzeit wie bei unbegrenztem Arbeitsspeicher berechenbar [9]. Im hier beschriebenen Fall treten fast keine Störungen auf, und damit erhält man praktisch bei begrenztem Arbeitsspeicher (2 Plätze) dasselbe Resultat, wie bei unbegrenztem Arbeitsspeicher, und zwar unabhängig vom Transport-Rechenzeitverhältnis $E(t_H)/E(t_R)$, solange es kleiner gleich 1,0 ist. Es gelingt also auch hier, Simulationsergebnisse durch ein berechenbares Modell nachzuvollziehen.

6. Überprüfung der gefundenen Arbeitsspeicher-Belegungsstrategien in einem realistischeren Modell

6.1 Simulationsmodell zur Bearbeitung von Aufgaben mit mehr als einer Teilaufgabe

Unter Berücksichtigung von Messungen im System des TR 440 muß das Modell verbessert werden: Aufgaben haben im Stapelbetrieb im Mittel mehr als 100 Teilaufgaben, und die Erwartungswerte der Transportzeiten für Laden und Verdrängen von Programmen (Bild 8) sind deutlich größer, als für Ergänzungstransporte, z. B.

$$E(T_L) = E(T_V) = 10\,E(t_E)\,. \qquad (6.1)$$

Wegen dieser sehr unterschiedlichen Mittelwerte ist eine Überprüfung der in Abschnitt 5.1 unter der Annahme gleichgroßer Mittelwerte $E(T_L) = E(T_V) = E(t_E)$ gefundenen Belegungsstrategien nötig. Es liegt der Verdacht nahe, daß die Annahme gleichgroßer Mittelwerte für Programm- und Ergänzungstransporte zu einer zu transportintensiven Arbeitsspeicher-Belegungsstrategie führt, denn das Verdrängen und Laden ganzer Programme wird scheinbar erleichtert.

Im Simulationsmodell wird die Aufgabenbearbeitung bei 2 Plätzen im Arbeitsspeicher nachgebildet (Bild 8). Unabhängig von der Anzahl der Plätze im Arbeitsspeicher zeigen Meßergebnisse (TR 440), daß nach Fertigbearbeitung einer Teilaufgabe durch den Rechnerkern mit konstanter Wahrscheinlichkeit eine weitere Teilaufgabe derselben Aufgabe nachfolgt. Setzt man sehr verschiedene Mittelwerte der Transportzeiten voraus (Gl. 6.1), so ergibt sich bei im Mittel 100 Teilaufgaben pro Aufgabe, daß nur bei großem Variationskoeffizienten der Teilaufgabenrechenzeit (z. B. VK = 10 in Gl. 2.2) die mittlere Antwortzeit durch die in Abschnitt 5.1 beschriebenen Verdrängungen verkleinert wird. Abhängig von der Benutzergemeinde eines Rechenzentrums kommen große und kleinere

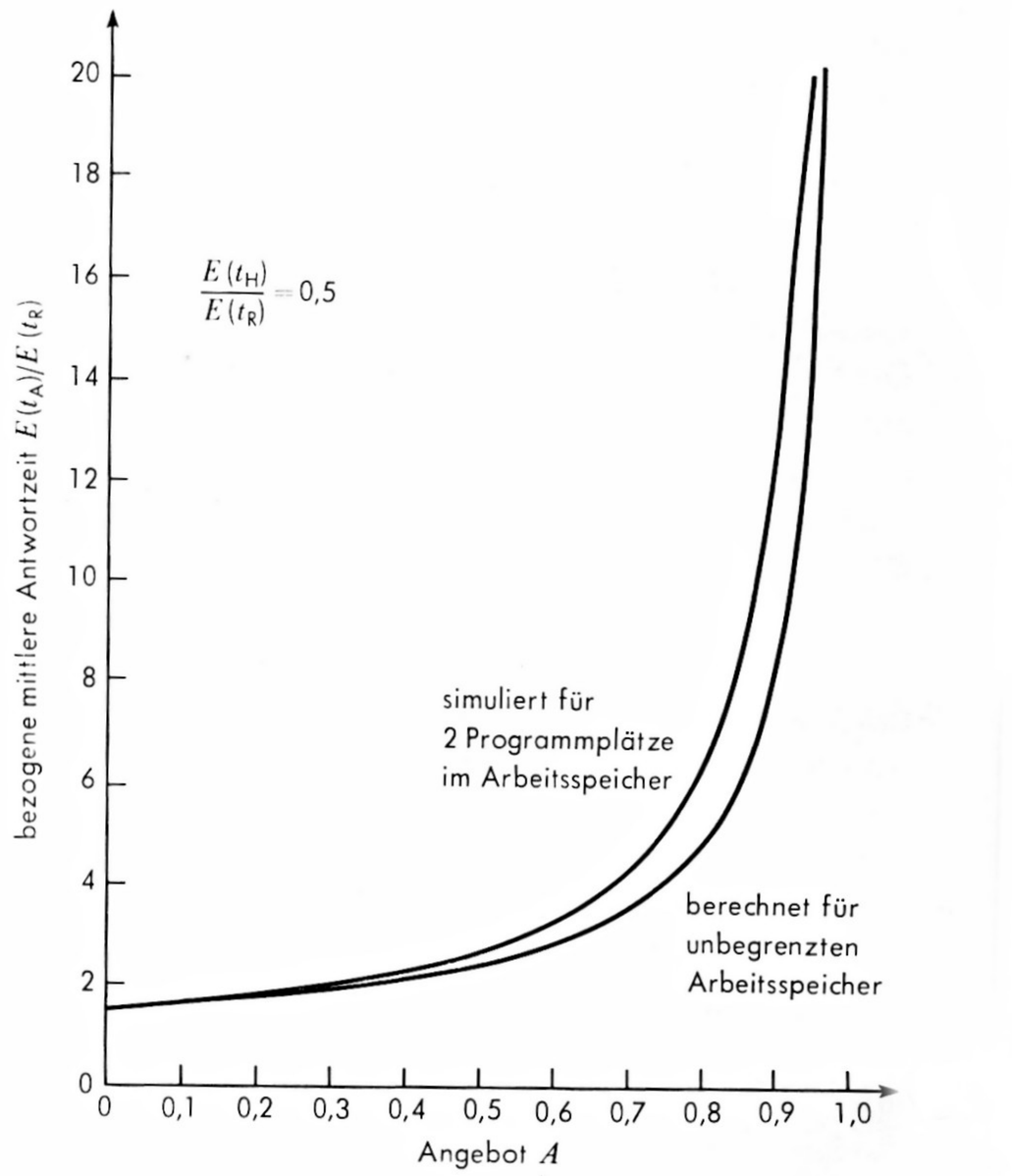

Bild 9. Bei stückweise exponentieller Rechenzeitverteilung (Bild 3) und optimaler Zeitzuteilung S1 ist ein begrenzt großer Arbeitsspeicher bei günstigster Platzzuteilung fast so gut wie ein unbegrenzter [8].

Variationskoeffizienten vor. Abhängig von der Hintergrundspeicher-Konfiguration und der Organisation des Betriebssystems unterscheiden sich die Mittelwerte für Programm- und Ergänzungstransporte um mehr oder weniger als den Faktor 10. Je nach Rechnerkonfiguration und Benutzerkreis wird man die unter 5.1 beschriebene verdrängende Strategie oder die Arbeitsspeicher-Belegungsstrategie FIFO (keine Verdrängungen) wählen [11, 13].

Für viele gemessene Teilaufgabenrechenzeitverteilungen ist der Variationskoeffizient deutlich kleiner als 10, so daß in vielen Betriebsfällen einmal in den Arbeitsspeicher geladene Programme bis zur Fertigstellung aller ihrer Teilaufgaben dort bleiben müssen. Man erhält dabei für Aufgaben ähnliche Ergebnisse, wie in Bild 9 für Teilaufgaben dargestellt. Anstelle der Erwartungswerte für Teilaufgaben sind solche für Aufgaben einzusetzen. Bemerkenswert ist, daß die Resultate für stückweise und entartet negativ exponentielle Rechenzeitverteilungen sehr nahe beieinander liegen, so daß man für die Berechnung die entartete Verteilung ansetzen darf.

6.2 Analytisches Modell für die Aufgabenbearbeitung bei begrenztem Arbeitsspeicher

Bei entartet negativ exponentiell verteilter Rechenzeit von Teilaufgaben ist die Rechnerkern-Zuteilungsstrategie S1 bei Verwaltungszeit $\tau = 0{,}0$ optimal und sonst praktisch optimal. Aufgrund von Simulationsergebnissen steht fest, daß bei 2 Plätzen im Arbeitsspeicher die Belegungsstrategie FIFO in vielen Fällen besonders günstig ist. Demnach kann man mit der belanglosen Vereinfachung $E(T_L) = E(t_E)$ anstelle des

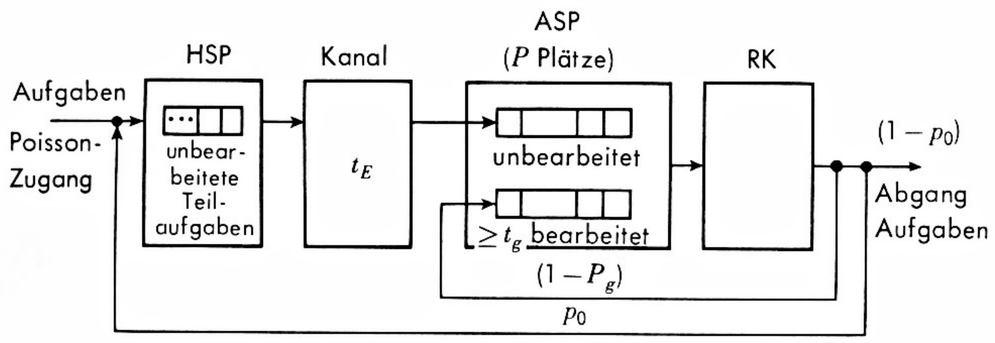

Bild 10. Analytisches Modell zur Berechnung der mittleren Antwortzeit von Aufgaben in einem Teilnehmersystem. Optimale RK-Zuteilung durch die Strategie S1, optimale Arbeitsspeicherzuteilung FIFO (keine Verdrängungen). Mit der Wahrscheinlichkeit p_0 folgt einer Teilaufgabe eine weitere. Im ASP sind P (=1, 2,...) Plätze für Programme vorhanden.

Modells nach Bild 8 ein analytisches Modell (Bild 10) angeben.

Die zwei Warteschlangen im ASP sind durch die Strategie S1 bedingt, die zwischen unbearbeiteten und Langrechnerteilaufgaben unterscheiden muß. Das Modell gilt für jede Arbeitsspeichergröße, denn wenn schon bei zwei Plätzen keine Verdrängungen notwendig bzw. zulässig sind, dann erst recht nicht bei mehr Plätzen. Unter dem Optimierungsziel kleinstmögliche mittlere Antwortzeit treten Verdrängungen, wenn überhaupt, nur wegen eines beengten Arbeitsspeichers auf.

Ein wichtiges Merkmal dieses Modells ist, daß die optimalen Strategien zur Zuteilung von RK und ASP bekannt sind. Bisher ist lediglich die Berechnung des Durchsatzes in Abhängigkeit von der Größe des ASP bei entartet negativ exponentieller Rechenzeitverteilung der Teilaufgaben bei P (=1, 2, 3...) Plätzen gelungen. Das Ergebnis ist an anderer Stelle veröffentlicht [12]. Es hat jedoch den Anschein, daß auch die Bestimmung der mittleren Antwortzeit möglich sein wird.

7. Zusammenfassung

Ausgehend von einem sehr einfachen Rechnermodell wird zunächst die Rechnerkernzuteilung bei Verwaltungszeit bei – durch einen unbegrenzt großen Arbeitsspeicher – entkoppelten Bedienstellen (Rechnerkern, Transportkanal) durch Simulation untersucht. Dabei zeigt sich, daß stückweise und entartet negativ exponentielle Verteilungen als Näherungen an gemessene Summenhäufigkeiten von Teilaufgabenrechenzeiten der Hyperexponentialfunktion als Näherung überlegen sind. Denn von diesen Verteilungen läßt sich, solange die mittlere Antwortzeit durch Unterbrechungen optimiert werden kann, eine von der Größe der Verwaltungszeit unabhängige, praktisch optimale Rechnerkern-Zuteilungsstrategie ableiten. Man kommt dabei zu einem berechenbaren Modell. Die Lösung des Teilproblems Rechnerkern-Zuteilung ist die Voraussetzung für die Suche nach optimalen Belegungsstrategien eines auf 2 Programmplätze begrenzten Arbeitsspeichers durch Simulation. Die bei günstigster Strategie gewonnenen Simulationsergebnisse werden durch eine Grenzbetrachtung näherungsweise berechenbar. Die Arbeit zeigt an Beispielen die wechselseitige Befruchtung von Simulation und analytischer Rechnung und gibt Hilfestellung bei der Modell-Konstruktion von Rechensystemen.

Literatur

[1] *Küspert, H.-J., Marte, G.*, (AEG-Telefunken), Optimale Rechenzeitzuteilung bei einem Teilnehmerrechensystem mit jeweils einer Aufgabe im Arbeitsspeicher. Elektron. Rechenanl. 12 (1970), H. 3, S. 155–162.

[2] *Marte, G.*, Optimal-time-scheduling for time-shared computer systems with piecewise exponential computing time distribution function. ACM Conference, Mai 1970, Bonn.

[3] *Marte, G.*, Zur Synthese von Teilnehmerrechensystemen. Wiss. Berichte AEG-Telefunken, 44 (1971), H. 3, S. 114–123.

[4] *McKinney, J. M.*, A survey of analytical time-sharing models. Computing Surveys, Vol. 1, No. 2, June 1969, S. 105–160.

[5] *Moll, William L.*, Measurement, analysis and simulation of computer center operations. AD-Bericht 711 293, Juni 1970, 132 Seiten.

[6] *Olivier, G.*, Optimale Zeitzuteilung für wartende Rechenaufgaben. Elektron. Rechenanl., 9 (1967), H. 5, S. 218–224.

[7] *Sakata, M., Noguchi, S., Oizumi, J.*, An analysis of the M/G/1 queue under round robin scheduling. Oper. Res. (USA), Vol 19, No. 2, 1971, S. 371–385.

[8] *Walke, B.*, Die mittlere Verweilzeit in Teilnehmerrechensystemen bei optimaler Rechenzeit-, Arbeitsspeicher- und Transportkanalzuteilung. Nachrichtentechn. Fachberichte, Bd. 44, 1972, VDE-Verlag Berlin, S. 257–264.

[9] *Walke, B., Küspert, H.-J.*, Teilnehmerrechensysteme: Mittlere Verweilzeiten bei optimaler Rechenzeitzuteilung. Elektron. Rechenanl. 13 (1971), H. 5, S. 193–199.

[10] Die Messungen wurden von *G. Mersmann*, TC Konstanz, durchgeführt.

[11] *Walke, B.*, Optimierung der Arbeitsspeicherbelegung in Teilnehmerrechensystemen durch Simulation. Wiss. Berichte AEG-TELEFUNKEN 47 (1974), H. 1.

[12] *Walke, B.*, Durchsatzberechnung für Modelle mit begrenztem Arbeitsspeicher bei einem und zwei Rechnerkernen, sowie einem und zwei Transportkanälen. Elektron. Rechenanl., 15 (1973), H. 5, S. 223—233.

[13] *Walke, B.*, Simulation und Rechnung bei der Durchsatzoptimierung für gemessene, technisch wissenschaftliche Aufgabenprofile. Lecture Notes in Economics and Mathematical Systems. NTG/GI-Fachtagung „Struktur und Betrieb von Rechenanlagen", Braunschweig, März 1974, Springer Berlin.