Warteschlangen
-
Proseminar Technische Informatik

(Andreas Weiß)

1. Einleitung

1.1 Überblick

Dieser Bericht befasst sich mit Warteschlangen, ihrer Funktionsweise und ihrem Nutzen. Wir werden uns mit den Markovschen Warteschlangen und den Nichtmarkovschen Warteschlangen auseinandersetzen. Des Weiteren gehen wir genauer auf die verschiedenen Möglichkeiten ein, wie eine Warteschlange konzipiert sein kann und beschäftigen uns mit den Grundlagen der Analyse eines Computersystems mithilfe einer Warteschlange. Außerdem sehen wir uns eine Fallstudie zum Thema Warteschlangen an.

1.2 Was ist eine Warteschlange?

Bevor wir uns genauer mit den technischen Funktionsweisen einer Warteschlange befassen, beantworten wir die Frage, was eine Warteschlange überhaupt ist. Eine Warteschlange dient dazu, bestimmte Aufträge oder Aufgaben in einer geordneten Reihenfolge abzuarbeiten. In einer Warteschlange werden diese Aufträge solange aufgeschoben, bis wieder eine entsprechende Ressource zur Verfügung steht, die den jeweiligen Auftrag bearbeiten kann. Ein einfaches Beispiel hierfür stellt eine Druckerwarteschlange dar.
Ein Drucker kann lediglich eine bestimmte Anzahl von Druckaufträgen pro Minute ausführen. Erhält der Drucker in einer Minute jedoch mehr Aufträge als er bearbeiten kann, müssen die entsprechenden Aufträge in einer Warteschlange abgelegt werden, bis die Ressource, in diesem Fall der Drucker, einen neuen Auftrag entgegennehmen kann. Ohne eine Warteschlange wäre jeder Nutzer dazu gezwungen, den Druckvorgang immer wieder in Auftrag zu geben.

1.3 Wofür werden Warteschlangen benötigt?

Abgesehen von dem obigen Beispiel gibt es sehr viele Verwendungsmethoden für Warteschlangen. Für diesen Bericht interessant ist speziell die Warteschlange zur Analyse von Computersystemen. In der IT-Branche lassen sich mithilfe von Warteschlangen Aussagen darüber treffen, wie effizient bestimmte Arbeitsvorgänge laufen oder wie hoch die Rechenleistung eines bestimmten Systems ist. Dafür werden Warteschlangen(-Netzwerke) ausgewählt, welche auf den zu vergleichenden Computersystemen ausgeführt werden. Je nachdem, wie schnell die jeweilige Warteschlange abgearbeitet wird, lässt sich feststellen, wie hoch die Leistung eines Systems im Vergleich zu genormten Leistungen ist.

1.4 Wie funktioniert eine Warteschlange?

Es gibt eine Reihe verschiedener Konzepte, nach denen eine Warteschlange arbeiten kann. Daher werden wir einen kurzen Blick auf die wichtigsten Funktionsarten werfen.

- First-Come-First-Served: Nach diesem Prinzip arbeiten die meisten Warteschlangen. Der Auftrag, welcher als erstes die Warteschlange betritt, wird als erstes bearbeitet.
- Last-Come-First-Served: Der Auftrag, welcher als letztes die Warteschlange betritt wird als erstes bearbeitet.
- Service –In-Random-Order: Es entscheidet der Zufall darüber, welcher Auftrag aus der Warteschlange als nächstes für eine Bearbeitung ausgewählt wird.
- Round Robin: Arbeitet prinzipiell nach dem FCFS-Konzept. Falls ein Auftrag jedoch eine zu lange Zeit in Anspruch nimmt, wird er unterbrochen und muss erneut in die Warteschlange. Dieser Vorgang wiederholt sich so lange, bis der betroffene Auftrag bearbeitet wurde.
- Processor Sharing: Nach diesem Prinzip arbeiten oftmals Download-Warteschlangen. Alle Aufträge werden (scheinbar) gleichzeitig bearbeitet. Jedoch erhöht sich mit jedem zusätzlichen Auftrag die Bearbeitungszeit des einzelnen Auftrags.
- Infinite Server: Es werden genügend Ressourcen zur Verfügung gestellt, sodass sich keine Warteschlange bilden kann.
- Static Priorities: Warteschlangen dieser Art sind nach Priorität geordnet. Ein wichtiger Auftrag wird entsprechend schnell bearbeitet (funktioniert natürlich auch anders herum, sofern gewollt). Haben mehrere Aufträge die gleiche Priorität, gilt das Prinzip FCFS.
- Dynamic Priorities: Die Priorität der eingehenden Aufträge hängt davon ab, wie lange die jeweilige Bearbeitung dauern wird. So können beispielsweise kleinere Aufträge zuerst abgearbeitet werden, um die Warteschlange möglichst übersichtlich zu halten.
- Preemption: Dies ist auch ein prioritätsorientiertes Prinzip einer Warteschlange. Es ist beinahe äquivalent zum Static Priorities - Prinzip, unterschiedet sich jedoch darin, dass eine Bearbeitung eines Auftrags jederzeit unterbrochen werden kann, sofern ein Auftrag höherer Priorität die Warteschlange betritt.

2. Technische Aspekte

2.1 Vorschau

Nachdem wir einen kurzen Überblick über die Definition und Funktion einer Warteschlange bekommen haben, werden wir uns mit diversen Warteschlangenmodellen, unter anderem den Markovschen Warteschlangen und den Nicht-Markovschen Warteschlangen, sowie dem Maschine-Mechaniker-Modell auseinandersetzen.

2.2 Die Notation von Kendall

Im Folgenden werden wir uns unterschiedliche Warteschlangenmodelle ansehen. Diese werden durch die Notation von Kendall spezifiziert. Dabei beinhalten die meisten Notationen drei Stellen. Die erste Stelle spezifiziert die Auftragsverteilung, die zweite gibt Auskunft über die Auftragsdauer und die dritte Stelle enthält Angaben über die Anzahl aller zur Verfügung stehenden Server. Dabei beruht Kendalls Notation auf verschiedenen Zeichen. Zum Beispiel:

- M: Exponentielle Verteilung
- D: Deterministische Verteilung
- G: Allgemeine Verteilung

Eine Kombination könne demnach folgendermaßen aussehen:
M / D / 30 → Die Menge der Aufträge verhält sich exponentiell, das heißt, es wird eine unbestimmte Anzahl an Aufträgen erwartet. Die Bearbeitungszeit ist hingegen deterministisch, was bedeutet, dass jeder Auftrag eine konstante Zeit in Anspruch nimmt. Unabhängig davon wird die Warteschlange von 30 Servern abgearbeitet.

2.3 Markovsche Warteschlangen

2.3.1 M / M / 1 – Modell

Beginnen wir mit dem M/M/1 – Modell. M/M/1 bedeutet: Sowohl die Menge der Aufträge als auch die Bearbeitungszeit sind exponentiell, das heißt, zwischen dem Eintreffen zweier Aufträge liegt ein zufällig großer Zeitraum. Analog dazu nimmt jeder Auftrag eine zufällige Bearbeitungszeit in Anspruch. Die Eins signalisiert uns, dass für die Abarbeitung der Aufträge lediglich ein Server zur Verfügung steht. Das M/M/1 – Modell basiert dabei auf folgenden Formeln ($\rho = \lambda / \mu$):
Wahrscheinlichkeit, dass sich Null Aufträge in der Warteschlange befinden:

$$\pi 0 = 1 - \lambda / \mu$$

Wahrscheinlichkeit, dass k Aufträge in der Warteschlange warten:

$$\pi k = (1 - \rho) * \rho^k$$

Durchschnittliche Anzahl an Aufträgen:

$$K = \rho / 1 - \rho$$

Durchschnittliche Wartezeit:

$$W = \rho / \mu / 1 - \rho$$

Durchschnittliche Warteschlangensystemlänge (Warteschlange + Server):

$$Q = \rho^2 \ / \ 1 - \rho$$

Zur besseren Veranschaulichung des M/M/1 – Modells übertragen wir diese Formeln auf eine Warteschlange aus dem Alltag (beispielsweise ein Laden). Wir wissen bereits, dass lediglich eine Kasse geöffnet ist (M/M/**1**) und gehen zunächst davon aus, dass kein großer Andrang besteht (→ λ ist ein unbestimmter kleiner Wert, z.B. 0.8). Des Weiteren nehmen wir an, dass die Bearbeitungszeit relativ gering ist, was eine hohe Bearbeitungsrate impliziert (→ μ ist ein unbestimmter großer Wert, z.B. 1.0). Nun setzen wir unsere beliebig ausgewählten Werte in die Formeln ein:

$\pi 0 = 1 - \lambda / \mu$

→ $\pi 0 = 1 – 0.8 / 1.0$

→ $\pi 0 = 1 – 0.8$

→ $\pi 0 = 0.2$

Die Wahrscheinlichkeit, dass kein Kunde warten muss, liegt demnach bei 20 %.

$\pi k = (1 - \rho) * \rho^k$

→ $\pi 5 = (1 – 0.8 / 1.0) * (0.8 / 1.0)^5$

→ $\pi 5 = 0.2 * 0.8^5$

→ $\pi 5 = 0.065536$

Die Wahrscheinlichkeit, dass 5 Kunden warten müssen, liegt bei 6.5536%.

Bei der Berechnung der Wahrscheinlichkeiten gilt folgende Regel: „Die Summe aller Wahrscheinlichkeiten von k = 0 bis k = unendlich für „πk“ muss 1 = 100 % betragen.“.

Eine entsprechende „Gegenprobe“ ist nicht möglich, das heißt, wir können die obige Formel nicht auf eine Ankunftsrate > Bearbeitungsrate übertragen, denn sonst würde die Warteschlange unendlich lang werden. Beweis:

Wir ändern unsere Annahmen:

→ $\lambda = 1.0$

→ $\mu = 0.8$

Durch das Einsetzen in die vorgegebenen Formeln erhalten wir folgende Ergebnisse:

$\pi 0 = 1 - \lambda / \mu$

→ $\pi 0 = 1 – 1.0 / 0.8$

→ $\pi 0 = 1 – 1.25$

→ $\pi 0 = -0.25$ (Ergebnis unbrauchbar)

2.3.2 M / M / ∞ – Modell

Das M/M/∞ – Modell arbeitet generell nach dem gleichen Prinzip, wie das M/M/1-Modell, mit dem Unterschied, dass in diesem Modell eine unendlich große Anzahl an Servern für die Auftragsbearbeitung bereitsteht. Dies impliziert, dass für jeden eintreffenden Auftrag immer ein Bearbeitungsplatz vorhanden ist. Rechnerisch stellt sich dies für uns folgendermaßen dar:

Wahrscheinlichkeit, dass sich k Aufträge in der Warteschlange befinden:

$\pi k = ((\lambda / \mu)^k / k!) * e ^ - (\lambda / \mu)$

Beispiel k=5 ; $\lambda = 10$; $\mu = 15$:

$\pi k = ((\lambda / \mu)^k / k!) * e ^ - (\lambda / \mu)$

→ $\pi 5 = ((10 / 15)^5 / 5!) * e ^ - (10 / 15)$

→ $\pi 5 = ((10 / 15)^5 / 5*4*3*2*1) * e ^ - (10 / 15)$

→ $\pi 5 = (0.00109739369) * e ^ - (10 / 15)$

→ $\pi 5 = 0.0005634207068 = 0.05634207068$ %

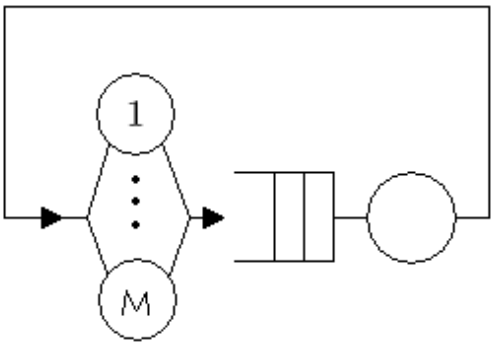
Zum Vergleich: Die Wahrscheinlichkeit unter denselben Voraussetzungen liegt im M/M/1 – Modell bei 0.043895747 = 4.3895747 %, womit die Wahrscheinlichkeit mehr als Siebenundsiebzigfache beträgt.

2.3.3 Nutzen für eine Analyse

Die Markovschen Warteschlangen dienen der Analyse. Mit ihnen kann bestimmt werden, ob eine vorhandene Serverkapazität ausreicht, um eine bestimmte oder unbestimmte Anzahl an Aufträgen zu bearbeiten. Dabei ist das M/M/1 – Modell das am besten geeignete von den beiden beschriebenen Markovschen Modellen. Denn das M/M/∞ - Modell ist praktisch nur selten umsetzbar. Aber auch das M/M/1 – Modell eignet sich nur bedingt für eine Analyse, denn in den wenigsten Fällen kann man von einem exponentiellen Eintreffen von Aufträgen ausgehen. Das M/M/1 – Modell berücksichtigt vor allem nicht die Uhrzeit.

2.4 Maschine – Mechaniker – Modell

Das Maschine – Mechaniker – Modell (englisch: Machine Repairman Model) ist ein relativ einfaches Modell zur Analyse eines Reparatur-Systems. Wir haben M Maschinen und einen Mechaniker. Beruhend auf der Größe von M, der Auftragsrate (in diesem Fall die Anzahl aller Maschinen, die in einem bestimmten Zeitraum kaputt gehen) und der Bearbeitungsrate (Arbeitsgeschwindigkeit des Mechanikers) lassen sich Aussagen darüber treffen, ob ein Reparatur-System im Alltag funktionieren würde. Kommt man dabei zu dem Schluss, dass der jeweilige Mechaniker überlastet ist, müssen mehr Mechaniker zur Verfügung gestellt werden.



Das Maschine – Mechaniker – Modell basiert auf den folgenden Formeln:

Wahrscheinlichkeit, dass k Maschinen auf eine Reparatur werten müssen:

$\pi k = \pi 0 * (\lambda / \mu)^k * M! / (M-k)!$, wobei

$\pi 0 = 1 /$ Summe von 0 bis M von $(\lambda / \mu)^k * M! / (M-k)!$

Durchschnittliche Wartezeit, bis eine Maschine wieder repariert ist:

$T = M / \mu(1 - \pi 0) – 1/\lambda$

Durchschnittliche Anzahl defekter Maschinen:

$K = M - \mu(1 - \pi 0) / \lambda$

Gehen wir beispielsweise davon aus, dass wir einen Mechaniker zur Verfügung haben, welcher die Wartung von 4 Maschinen überwachen soll. Wir gehen davon aus, dass der Mechaniker nur eine Maschine pro Stunde reparieren kann und durchschnittlich 0.8 Maschinen pro Stunde eine Reparatur benötigen. Folglich errechnen wir die durchschnittliche Wartezeit einer kaputten Maschine bis zur Reparatur durch:

$T = M / \mu(1 - \pi 0) – 1/\lambda$

➔ $T = 4 / 1.0(1 - \pi 0) – 1/0.8$

➔ $T = 4 / (1 – 0.000126758778) – 1/0.8$

➔ $T = 2.750504064$

In diesem Fall beträgt die Wartezeit durchschnittlich 2.75 Stunden, bis eine defekte Maschine wieder betriebsbereit ist. Diese vergleichsweise hohe Dauer ist dadurch zu erklären, dass die Wahrscheinlichkeit sehr hoch ist, dass mehrere Maschinen zeitgleich einen Defekt haben. So liegt die Wahrscheinlichkeit für k = 3 beispielsweise bei ungefähr 19 %.

2.5 Nichtmarkovsche Warteschlangen

Abgesehen von den Markovschen Warteschlangen gibt es eine Vielzahl anderer Modelle, die die Analyse von Warteschlangensystem ermöglichen. Im Folgenden setzen wir uns mit dem M/G/1-Modell sowie dem GI/M/m-Modell genauer auseinander.

2.5.1 M / G / 1 - Modell

Das M/G/1 – Modell arbeitet nach dem FCFS – Prinzip. Es ist deutlich komplexer als das Markovsche M/M/1-Modell, basiert jedoch weitgehend auf denselben Grundlagen. Die Auftragsrate ist exponentiell gewählt, die Bearbeitungsrate allgemein (im Prinzip ähnlich der exponentiellen Verteilung). Wie bereits im M/M/1 – Modell gibt es auch hier lediglich eine Bearbeitungsstation.

Gehen wir für einen direkten Vergleich von den gleichen Bedingungen aus, die wir bereits im M/M/1 – Beispiel gewählt haben. Dies impliziert, wir gehen von einer Auftragsrate von 0.8 und einer Bearbeitungsrate von ungefähr 1.0 aus. Im Folgenden errechnen wir die durchschnittliche Warteschlangensystemlänge Q, welche durch die Pollaczek-Khintchine-Formel:

$$Q = (\rho^2 \ / \ (1 - \rho)) \ * \ ((1 + cb^2) / 2) \quad \rightarrow \ (cb^2(\text{exponentiell}) = 1)$$

errechnet wird. Unter den nichtmarkovschen Warteschlangen hat ρ zudem eine andere Bedeutung und beschreibt nun $\lambda * E[S]$, wobei wir $E[S]$ als einen gegebenen Wert betrachten können. Wir gehen von einem Wert 0.9 aus.

- ➔ $Q = ((0.8 * 0.9)^2 \ / \ (1 - (0.8 * 0.9))) \ * \ ((1 + 1) / 2)$
- ➔ $Q = 1.851428571$

Die Durchschnittswartezeit pro Auftrag errechnen wir durch eine etwas komplexere Formel:

	W	= W0 + Q * TB	, wobei
	W0	= ρ * R	, wobei
	R	= E[S] / 2 * (1+cs^2)	(cs = Variationskoeffizient = 1)
→	R	= 0.9 / 2 * 2 → 0.9	
→	W0	= 0.72 * 0.9	
→	W	= 0.648 + 1.851428571 * TB	, wobei
	TB	= 1.0 (TB sei frei gewählt, in diesem Falle 1.0)	
→	W	= 0.9 + 1.851428571 * 1.0	
→	W	= 2.499428571	

Zum Vergleich (M/M/1):

$W = \rho / \mu / 1 - \rho$

- ➔ $W = 4$

2.5.2 M / G / m - Modell

Die Notation des M/G/m – Modells bedeutet, dass es eine exponentielle Anzahl an Aufträgen gibt, welche einen zufällig langen Zeitraum beanspruchen (Bearbeitungsrate = allgemein), dabei jedoch von einer unbestimmten Menge m an Servern bearbeitet werden.

Zum direkten Vergleich mit den anderen vorgestellten Warteschlangensystemen werden wir erneut die durchschnittliche Wartezeit im gegebenen Modell berechnen. Dafür stellt uns das M/G/m – Modell die folgenden Formeln zur Verfügung:

Beispiel: $\lambda = 0.8$, $\mu = 1.0$, $m = 5$

	W	$= W0 + (Q / m) * T$	, wobei
	T	$= 1.0$ (T sei frei gewählt, in diesem Falle 1.0)	und
	$W0$	$= Pm * R$	, wobei
	Pm	$= (\rho^m + \rho) / 2$, wenn $\rho > 0.7$	, bzw.
		$= \rho \wedge ((m+1) / 2)$, wenn $\rho < 0.7$	, wobei
	ρ	$= \lambda * E[S]$	
→	ρ	$= 0.72$	
→	Pm	$= (0.72^5 + 0.72) / 2$	
→	Pm	$= 0.456745881$	und
	R	$= T * ((1 + cb^2) / 2m)$	
→	R	$= 1.0 * ((1 + 1^2) / 2*5)$	
→	R	$= 0.2$	
→	$W0$	$= 0.456745881 * 0.2 = 0.091349176$	
→	W	$= 0.46163489$ (für $Q = 1.851428571$)	

Das Ergebnis ist demnach nahezu ein Fünftel der durchschnittlichen Wartezeit im M/G/1 – Modell.

2.6 Wiederaufnahme

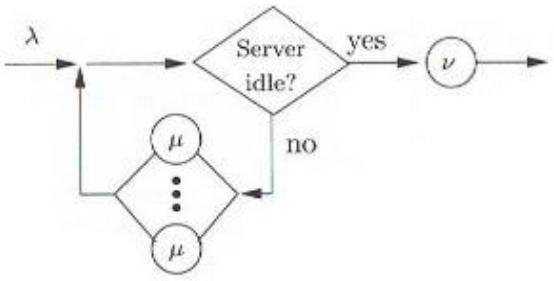
2.6.1 Wiederaufnahme von Aufträgen

Eine andere Form der Warteschlange stellt die Wiederaufnahme-Warteschlange dar. Entgegen der Vorgehensweise bei einer normalen Warteschlange bildet sich in einer Wiederaufnahme-Warteschlange lediglich eine Warteschleife. Diese funktioniert nach dem folgenden Prinzip:

Es gibt eine feste Anzahl an Servern (n) und, wie in den bisherigen Modellen, eine Auftragsrate (λ). Wenn ein Auftrag die Warteschlange betritt, wird zunächst überprüft, ob einer der n Server für die Bearbeitung zur Verfügung steht. Ist ein Server frei, wird der Auftrag bearbeitet. Ist kein Server frei, betritt der Auftrag eine Warteschleife. In zufälligen Zeitabständen werden erneut Bearbeitungsanfragen an das System gesendet. Dieser Vorgang wiederholt sich so oft, bis der entsprechende Auftrag bearbeitet wurde.

In dem Buch „Queueing Networks and Markov Chains“ wird die Wiederaufnahme-Warteschlange auf Beispiele aus dem Alltag übertragen. Besonders treffend ist hierbei der Vergleich zu einem Telefonat, denn dieses arbeitet nach demselben Prinzip: Zuerst versucht der User, in diesem Fall der Anrufende, einen Server, also jemand anderen, zu erreichen. Ist dieser jemand nicht erreichbar, ertönt das Besetztzeichen. In diesem Fall betritt der Anrufende keine Warteschlange, sondern wartet mehrere Minuten, ehe erneut versucht wird, den Gesprächspartner zu erreichen.

Im Falle des Alltags hat dieses Warteschlangensystem den entscheidenden Vorteil, dass ein User keine Kapazitäten in einer Warteschlange verschwendet, sondern während des Wartens andere Aufgaben erledigen kann. Allerdings berücksichtigt das Modell der Wiederaufnahme-Warteschlange nicht die Reihenfolge, in welcher die ersten Anfragen der User gestellt wurden. Im Zweifelsfall ist die Telefonleitung nach mehreren Minuten erneut besetzt, da sich inzwischen eine dritte Person „vorgedrängelt“ hat.



(Quelle: Queuing Networks and Markov Chains)

2.6.2 M / M / 1 - Wiederaufnahme

Übertragen wir nun das Prinzip der Wiederaufnahme-Warteschlange auf das Markovsche M/M/1-Modell. Zur Erinnerung: Das M/M/1 – Modell geht von einer exponentiellen Auftragsrate, sowie einer exponentiellen Bearbeitungsrate und einem Server aus. Nun gehen wir von einem zusätzlichen Element (v) aus, welches eine exponentielle Wiederaufnahmerate beschreibt. Seien zudem C und N die Anzahl aller Systemanfragen, welche erneut gestellt werden müssen, wobei C die Anzahl jener Anfragen im Server und N die Anzahl jener Anfragen in Warteschleifen beschreiben. Dann ist:

P(i,n)	=	P[C=i, N=n]	, wobei
i	=	[0,1]	und
n	$\geq$	0	

P(1,8) bedeutet demnach: Es befindet sich eine erneute Anfrage im Server und acht weitere in jeweiligen Warteschleifen:

P(0,n)	=	ρ^n / n! * v^n * P(0,0)*Produkt(i=0 bis n-1) (λ + iv)	mit n $\geq$ 1
P(1,n)	=	ρ^(n+1) / n! * v^n * P(0,0)*Produkt(i=1 bis n) (λ + iv)	mit n $\geq$ 0
P(0,0)	=	(1-ρ) ^ (λ/v + 1)	, wobei
ρ = λ/μ			

Zudem erhalten wir durch die folgenden Rechnungen entsprechende Werte:

Durchschnittliche Anzahl aller Wiederaufnahmeanfragen im System:

K = ρ / (1-ρ) * (1 + λ/v)

Durchschnittliche „Warteschlangenlänge“:

Q = ρ^2 / (1-ρ) * (1 + μ/v)

Durchschnittliche Rückmeldungszeit:

T = 1/μ * 1 / (1-ρ) * (1 + λ/v)

Durchschnittliche Wartezeit:

W = 1/μ * ρ / (1-ρ) * (1 + μ/v)

Für die durchschnittliche „Warteschlangenlänge“ in einer Wiederaufnahme-Warteschlange (Q) ergeben sich in Abhängigkeit zur Wiederaufnahmerate (v) folgende Werte: (Quelle: Queueing Networks and Markov Chains)

v	0.0	0.2	0.5	1.0	1.5	2.0	5.0	10.0	∞
Q	∞	19.2	9.60	6.40	5.30	4.80	3.82	3.52	3.20

2.7 Fallstudie

Die folgende Fallstudie zum Thema Warteschlangen wurde am 07.01.2010 an der Freien Universität Berlin zwischen 11.30 und 14.00 Uhr durchgeführt. Sie befasst sich mit der Auslastung der Drucker im Informatik-Institut.

Bevor wir die Ergebnisse auswerten, werden wir die vorliegende Warteschlange durch Kendalls Notation genauer spezifizieren. Es liegt eine M/M/2 – Warteschlange vor. Wir haben eine exponentielle Verteilung an Aufträgen, die jeweils eine exponentielle Zeit in Anspruch nehmen. Dabei werden dem System zwei Server, in diesem Fall Drucker, zur Verfügung gestellt. Demnach basiert diese Warteschlange auf dem Markovschen M/M/m – Modell, wobei m gleich zwei gewählt wird. Bei der Studie wurde die Länge der Warteschlange für beide Drucker in einem Zeitabstand von jeweils zehn Minuten gemessen.

Uhrzeit	Warteschlangenlänge	Zeit für Druck (1 Blatt)
11.30	0	10 Sekunden
11.40	0	
11.50	0	
12.00	3	
12.10	2	
12.20	3	1 Minute
12.30	1	
12.40	0	
12.50	2	20 Sekunden
13.00	2	
13.10	1	
13.20	1	20 Sekunden
13.30	0	
13.40	1	
13.50	0	
14.00	0	

Anhand dieser Ergebnisse berechnen wir im Folgenden die Wahrscheinlichkeit, dass sich höchstens fünf Aufträge in der Warteschlange befinden, sowie die Wahrscheinlichkeit, dass ein eintreffender Auftrag generell zu Warten hat als auch die durchschnittliche Warteschlangenlänge. Dabei Beträgt die Auftragsrate durchschnittlich 1.0 Aufträge pro Minute, die Bearbeitungsrate ungefähr 5.0 Aufträge pro Minute. Die Bearbeitungsrate beruht hierbei auf Beobachtungen im Zeitraum von jeweils einer Minute und berücksichtigt keine Ausnahmen, wie beispielsweise größere Druckaufträge oder aufgebrauchtes Papier.

Wahrscheinlichkeit, dass sich höchstens k = 5 Aufträge in der Warteschlange befinden:

$\pi k = \pi 0 * ((\rho^k * m^m) / m!)$ für $k > m$

$= \pi 0 * ((m\rho)^k / k!)$ für $0 < k \leq m$, wobei

$\rho = \lambda / (m\mu)$ und

	$\pi 0$	=	(Summe(k=0 bis m-1) von (mρ)^k/k! + (mρ)^m/m! * 1/(1-ρ))^(-1)		
→	$\pi 0$	=	(Summe(k=0 bis 1) von (0.2)^k/k! + (0.2)^2/2 * 1/(0.9))^(-1)		
		=	0.803571428 = 80.3571428 %		
→	$\pi 5$	=	(0.1^5 * 2^2 / 2) * 0.803571428	=	0.000016
→	$\pi 4$	=	(0.1^4 * 2^2 / 2) * 0.803571428	=	0.00016
→	$\pi 3$	=	(0.1^3 * 2^2 / 2) * 0.803571428	=	0.0016
→	$\pi 2$	=	(0.2^2 / 2) * 0.803571428	=	0.016
→	$\pi 1$	=	(0.2^1 / 1) * 0.803571428	=	0.16

Die Wahrscheinlichkeit, dass höchstens 5 Kunden warten müssen, liegt demnach bei: 0.000016 + 0.00016 + 0.0016 + 0.016 + 0.16 + 0.803 = 0.980776, also 98 %. Die Wahrscheinlichkeit, dass ein eintreffender Auftrag generell in einer Schlange warten muss, berechnen wir folgendermaßen:

	Pm	=	((mρ)^m / (m!(1 - ρ))) * $\pi 0$
→	P2	=	((0.2)^2 / (2!(0.9))) * 0.8
		=	0.0144 = 1.44 %

Daraus berechnen wir im Folgenden die durchschnittliche Warteschlangenlänge Q:

	Q	=	ρ / (1 - ρ) * Pm
→	Q	=	(0.1 / 0.9) * 0.0144 = 0.0016

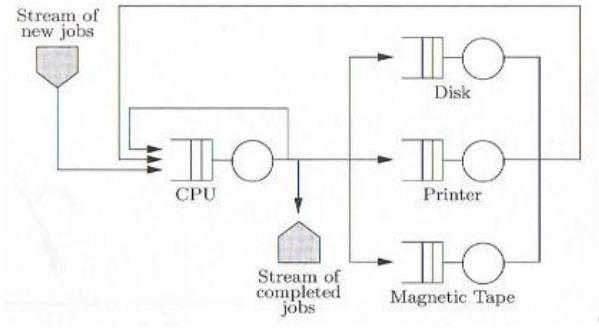
Der Grund für die beiden kleinen Werte ist der relativ geringe Zeitraum, welcher die Studie umfasst, sowie das Zusammenfassen von verschiedenen Auftragsraten aus Zeiträumen mit einer geringen Auslastung mit anderen aus einem Zeitraum mit höherer Druckerauslastung.

Anhand des M/M/m – Modells haben wir die Effizienz des Druckersystems im Informatik-Institut bewertet und kommen zu dem Ergebnis, dass zwei Drucker ausreichen, um die Aufträge aller Benutzer innerhalb eines kurzen Zeitraums zu bearbeiten, da die Warteschlange in 98% aller Fälle kleiner gleich 5 bleibt.

3. Warteschlangennetzwerke

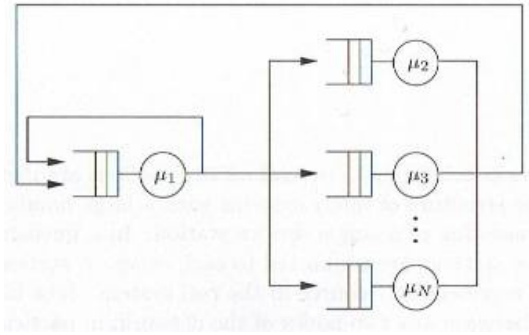
3.1 Was ist ein Warteschlangennetzwerk

Ein Warteschlangennetzwerk funktioniert generell wie eine einzelne Warteschlange, mit dem Unterschied, dass hier mehrere Warteschlangensysteme miteinander verknüpft sind. Ein System kann dabei aus einem oder mehr Knoten bestehen. Ein eintreffender Auftrag kann zwischen allen Systemen und allen Knoten wechseln und somit mehrere Bearbeitungsstationen durchlaufen. Dabei gibt es zwei Funktionsarten von Warteschlangennetzwerken. Zum Einen gibt es das „offene Netzwerk“. In ihm können beliebig viele Aufträge an einer beliebigen Stelle des Netzwerkes eintreten und ebenfalls an einer beliebigen Stelle wieder verlassen.



(Quelle: Queuing Networks and Markov Chains)

Dem gegenüber steht das „geschlossene Netzwerk“, welches nur eine bestimmte Anzahl an Aufträgen aufnehmen kann.



(Quelle: Queuing Networks and Markov Chains)

Wählen wir einen Onlineshop als Alltagsbeispiel. In diesem Fall handelt es sich um ein geschlossenes System. Der Kunde kann lediglich über die Bestellfunktion das Warteschlangennetzwerk betreten und die Kapazität an Lieferungen ist begrenzt. Zu Beginn schickt der Kunde eine Bestellung ab. Diese durchläuft nun mehrere Prozesse. Die Bestellung muss registriert, verarbeitet und geliefert werden. All diese Prozesse bestehen aus Warteschlangen, welche zu einem großen Netzwerk zusammengefasst werden.

3.2 Typen

3.2.1 Warteschlangennetzwerke innerhalb eines Systems

Hierbei handelt es sich um ein „einzelnes“ Warteschlangennetzwerk, beispielsweise der oben genannte Onlineshop. Für uns ist speziell der Aspekt der Analyse interessant, denn wie bereits bei den Markovschen und Nichtmarkovschen Warteschlangen lassen sich auch im Warteschlangennetzwerk bestimmte Teilelemente berechnen:

λ	=	Allgemeine Auftragsrate
λi	=	Auftragsrate am Knoten i
$\lambda 0i$	=	Auftragsrate von außen an Knoten i
Pij	=	Wahrscheinlichkeit, dass Auftrag von i zu j übergeht

Dabei berechnen wir λi folgendermaßen: Als Beispiel wählen wir ein System mit drei Knoten, wobei $\lambda 1$ = unbekannt, $\lambda 0i = 0.8$, $\lambda 2 = 1.0$, $\lambda 3 = 1.5$ und Pj1 = 1/3 (33% zu Knoten 1, 33% zu Knoten 2 bzw. 3 und 33% aus dem Netzwerk hinaus)

λi = $\lambda 0i$ + Summe(j=1…N(Knotenanzahl)) von λj * Pji

$\lambda 1$ = 0.8 + (1.0 * 1/3 + 1.5 * 1/3)

$\lambda 1$ = 1.633333…

Im Folgenden lässt sich ebenfalls die Rate aller Bearbeitungen durch Knoten i berechnen:

ei = $\lambda i / \lambda$

e1 = 1.633333… / 2.0 (für $\lambda = 2.0$)

e1 = 0.8166666…

3.2.2 Warteschlangennetzwerke mit anderen Systemen

In diesem Fall kann man zwar ebenfalls von einem Warteschlangennetzwerk reden, jedoch ist dieses wesentlich komplexer als das Warteschlangennetzwerk innerhalb eines Systems. Speziell hier gibt es eine weitere Funktionsart von Warteschlangennetzwerken, das „vermischte Netzwerk“. Ein Warteschlangennetzwerk wird genau dann als Solches bezeichnet, wenn es sowohl offene als auch geschlossene Netzwerke im Gesamtnetzwerk gibt. Zur Analyse betrachten wir nun die dazugehörenden Funktionen:

Pir,0 = Wahrscheinlichkeit, dass ein Auftrag das Netzwerk verlässt, nachdem er im r-ten System im i-ten Knoten bearbeitet wurde.

= 1 – Summe(j=1…N) Summe(s=1…R) Pir,js

λir = Auftragsrate am i-ten Knoten im r-ten System.

= λ * P0,ir + Summe(j=1…N) Summe(s=1…R) λjs * Pjs,ir

eir = Bearbeitungsrate des i-ten Knotens im r-ten System

= P0,ir + Summe(j=1…N) Summe(s=1…R) ejs * Pjs,ir (offen)

= Summe(j=1…N) Summe(s=1…R) ejs * Pjs,ir (geschlossen)

4. Literaturangabe

1. Performance of Computer Communication Systems
   (A Model-Based Approach)
   Boudewijn R. Haverkort
2. Queuing Networks and Markov Chains
   (Modeling and Performance Evaluation with Computer Science Applications)
   (Second Edition)
   Gunter Bolch
   Stefan Greiner
   Hermann de Meer
   Kishor S. Trivedi