PROCEEDINGS / KONGRESOPSOMMINGS

6th
SOUTHERN AFRICAN COMPUTER SYMPOSIUM

6de
SUIDELIKE-AFRIKAANSE REKENAARSIMPOSIUM

De Overberger Hotel, Caledon
2 - 3 JULY 1991

SPONSORED by
ISM
FRD
GENMIN

EDITED by
M H LINCK
Department of Computer Science
University of Cape Town
TABLE OF CONTENTS

Foreword 1

Organising Committee 2

Referees 3

Program 5

Papers (In order of presentation) 9

"A value can belong to many types"
B H Venter, University of Fort Hare 10

"A Transputer Based Embedded Controller Development System"
M R Webster, R G Harley, D C Levy & D R Woodward,
University of Natal 16

"Improving a Control and Sequencing Language"
G Smit & C Fair, University of Cape Town 25

"Design of an Object Orientated Framework for Optimistic
Parallel Simulation on Shared-Memory Computers"
P Machanick, University of Witwatersrand 40

"Using Statecharts to Design and Specify the GMA
Direct-Manipulation User Interface"
L van Zijl & D Mitton, University of Stellenbosch 51

"Product Form Solutions for Multiserver Centres
with Hierarchical Classes of Customers"
A Krzesinski, University of Stellenbosch and
R Schassberger, Technische Universität Braunschweig 69

"A Reusable Kernel for the Development of Control Software"
W Fouche and P de Villiers, University of Stellenbosch 83

"An Implementation of Linda Tuple Space
under the Helios Operating System"
P G Clayton, E P Wentworth, G C Wells and F de Heer-Menlah,
Rhodes University 95

"The Design and Analysis of Distributed Virtual Memory
Consistency Protocols in an Object Orientated Operating System"
K Macgregor, University of Cape Town & R Campbell University
of Illinois at Urbana-Champaign 107
"Concurrency Control Mechanisms for Multidatabase Systems"
A Deacon, University of Stellenbosch
118

"Extending Local Recovery Techniques for Distributed Databases"
H L Victor & M H Rennhackkamp, University of Stellenbosch
135

"Analysing Routing Strategies in Sporadic Networks"
S Melville, University of Natal
148

The Design of a Speech Synthesis System for Afrikaans"
M J Wagener, University of Port Elizabeth
167

"Expert Systems for Management Control: A Multiexpert Architecture"
V Ram, University of Natal
177

"Integrating Similarity-Based and Explanation-Based Learning"
G D Oosthuizen and C Avenant, University of Pretoria
187

"Efficient Evaluation of Regular Path Programs"
P Wood, University of Cape Town
201

"Object Orientation in Relational Databases"
M Rennhackkamp, University of Stellenbosch
211

"Building a secure database using self-protecting objects"
M Olivier and S H von Solms, Rand Afrikaans University
228

"Modelling the Algebra of Weakest Preconditions"
C Brink and I Rewitsky, University of Cape Town
242

"A Model Checker for Transition Systems"
P de Villiers, University of Stellenbosch
262

"A New Algorithm for Finding an Upper Bound of the Genus of a Graph"
D I Carson and O R Oellermann, University of Natal
276
FOREWORD

The 6th Computer Symposium, organised under the auspices of SAICS, carries on the tradition of providing an opportunity for the South African scientific computing community to present research material to their peers.

It was heartening that 31 papers were offered for consideration. As before all these papers were refereed. Thereafter a selection committee chose 21 for presentation at the Symposium.

Several new dimensions are present in the 1991 symposium:

* The Symposium has been arranged for the day immediately after the SACLA conference.

* It is being run over only 1 day in contrast to the 2-3 days of previous symposia.

* I believe that it is first time that a Symposium has been held outside of the Transvaal.

* Over 85 people will be attending. Nearly all will have attended both events.

* A Sponsorship package for both SACLA and the Research Symposium was obtained. (This led to reduced hotel costs compared to previous symposia)

A major expense is the production of the Proceedings of the Symposium. To ensure financial soundness authors have had to pay the page charge of R20 per page.

A thought for the future would be consideration of a poster session at the Symposium. This could provide an alternative approach to presenting ideas or work.

I would sincerely hope that the twinning of SACLA and the Research Symposium is considered successful enough for this combination survive. As to whether a Research Symposium should be run each year after SACLA, or only every second year, is a matter of need and taste.

A challenge for the future is to encourage an even greater number of MSc & PhD students to attend the Symposium. Unlike this year, I would recommend that they be accommodated at the same cost as everyone else. Only if it is financially necessary should the sponsored number of students be limited.

I would like to thank the other members of the organising committee and my colleagues at UCT for all the help that they have given me. A special word of thanks goes to Prof. Pieter Kritzinger who has provided me with invaluable help and ideas throughout the organisation of this 6th Research Symposium.

M H Linck
Symposium Chairman
SYMPOSIUM CHAIRMAN

M H Linck, University of Cape Town

ORGANISING COMMITTEE

D Kourie, Pretoria University.
P S Kritzinger, University of Cape Town.
M H Linck, University of Cape Town.

SPONSORS

ISM
GENMIN
FRD
LIST OF REFEREES FOR 6th RESEARCH SYMPOSIUM

<table>
<thead>
<tr>
<th>NAME</th>
<th>INSTITUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barnard, E</td>
<td>Pretoria</td>
</tr>
<tr>
<td>Becker, Ronnie</td>
<td>UCT</td>
</tr>
<tr>
<td>Berman S</td>
<td>UCT</td>
</tr>
<tr>
<td>Bishop, Judy</td>
<td>Wits</td>
</tr>
<tr>
<td>Berman, Sonia</td>
<td>UCT</td>
</tr>
<tr>
<td>Brink, Chris</td>
<td>UCT</td>
</tr>
<tr>
<td>Bodde, Ryn</td>
<td>Networks Systems</td>
</tr>
<tr>
<td>Bornman, Chris</td>
<td>UNISA</td>
</tr>
<tr>
<td>Bruwer, Piet</td>
<td>UOFS</td>
</tr>
<tr>
<td>Cherenack, Paul</td>
<td>UCT</td>
</tr>
<tr>
<td>Cook Donald</td>
<td>UCT</td>
</tr>
<tr>
<td>de Jaeger, Gerhard</td>
<td>UCT</td>
</tr>
<tr>
<td>de Villiers, Pieter</td>
<td>Stellenbosch</td>
</tr>
<tr>
<td>Ehlers, Elize</td>
<td>RAU</td>
</tr>
<tr>
<td>Eloff, Jan</td>
<td>RAU</td>
</tr>
<tr>
<td>Finnie, Gavin</td>
<td>Natal</td>
</tr>
<tr>
<td>Gaynor, N</td>
<td>AECI</td>
</tr>
<tr>
<td>Hutchinson, Andrew</td>
<td>UCT</td>
</tr>
<tr>
<td>Jourdan, D</td>
<td>Pretoria</td>
</tr>
<tr>
<td>Kourie Derrick</td>
<td>Pretoria</td>
</tr>
<tr>
<td>Kritzinger, Pieter</td>
<td>UCT</td>
</tr>
<tr>
<td>Krzesinski, Tony</td>
<td>Stellenbosch</td>
</tr>
<tr>
<td>Laing, Doug</td>
<td>ISM</td>
</tr>
<tr>
<td>Labuschagne, Willem</td>
<td>UNISA</td>
</tr>
<tr>
<td>Levy, Dave</td>
<td>Natal</td>
</tr>
<tr>
<td>Name</td>
<td>Institution</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>MacGregor, Ken</td>
<td>UCT</td>
</tr>
<tr>
<td>Machanick, Philip</td>
<td>Wits</td>
</tr>
<tr>
<td>Mattison Keith</td>
<td>UCT</td>
</tr>
<tr>
<td>Messerschmidt, Hans</td>
<td>UOFS</td>
</tr>
<tr>
<td>Mutch, Laurie</td>
<td>Shell</td>
</tr>
<tr>
<td>Neishlos, N</td>
<td>Wits</td>
</tr>
<tr>
<td>Oosthuizen, Deon</td>
<td>Pretoria</td>
</tr>
<tr>
<td>Peters Joseph</td>
<td>Simon Fraser</td>
</tr>
<tr>
<td>Ram, V</td>
<td>Natal, Pmb.</td>
</tr>
<tr>
<td>Postma, Stef</td>
<td>Natal, Pmb</td>
</tr>
<tr>
<td>Rennhackkamp, Martin</td>
<td>Stellenbòsch</td>
</tr>
<tr>
<td>Shochot, John</td>
<td>Wits</td>
</tr>
<tr>
<td>Silverberg, Roger</td>
<td>Council for Mineral Technology</td>
</tr>
<tr>
<td>Smit, Riel</td>
<td>UCT</td>
</tr>
<tr>
<td>Smith, Dereck</td>
<td>UCT</td>
</tr>
<tr>
<td>Terry, Pat</td>
<td>Rhodes</td>
</tr>
<tr>
<td>van den Heever, Roelf</td>
<td>UP</td>
</tr>
<tr>
<td>van Zijl, Lynette</td>
<td>Stellenbosch</td>
</tr>
<tr>
<td>Venter, Herman</td>
<td>Fort Hare</td>
</tr>
<tr>
<td>Victor, Hema</td>
<td>Stellenbosch</td>
</tr>
<tr>
<td>von Solms, Basie</td>
<td>RAU</td>
</tr>
<tr>
<td>Wagenaar, M</td>
<td>UPE</td>
</tr>
<tr>
<td>Wentworth, Peter</td>
<td>Rhodes</td>
</tr>
<tr>
<td>Wheeler, Graham</td>
<td>UCT</td>
</tr>
<tr>
<td>Wood, Peter</td>
<td>UCT</td>
</tr>
</tbody>
</table>
6TH RESEARCH SYMPOSIUM - 1991
FINAL PROGRAM

TUESDAY 2nd July 1991

10h00 - 13h00 Registration
13h00 - 13h50 PUB LUNCH

14h00 - 15h30 SESSION 1A
Venue: Hassner
Chairman: Prof Basie von Solms

14h00 - 14h30
"A value can belong to many types."
B H Venter, University of Fort Hare

14h30 - 15h00
"A Transputer Based Embedded Controller Development System"
M R Webster, R G Harley, D C Levy & D R Woodward, University of Natal

15h00 - 15h30
"Improving a Control and Sequencing Language"
G Smit and C Fair, University of Cape Town

15h30 - 16h00 TEA

SESSION 1B
Venue: Hassner C
Chairman: Prof Roelf v d Heever

14h00 - 14h30
"Design of an Object Orientated Framework for Optimistic Parallel Simulation on Shared-Memory Computers"
P Machanick, University of Witwatersrand

14h30 - 15h00
"Using Statecharts to Design and Specify the GMA Direct-Manipulation User Interface"
L van Zijl & D Mitton, University of Stellenbosch

15h00 - 15h30
"Product Form Solutions for Multiserver Centres with Hierarchical Classes of Customers"
A Krzesinski, University of Stellenbosch and R Schassberger,
Technische Universität Braunschweig
16h00 - 17h30 SESSION 2A

Venue: Hassner

Chairman: Prof Derrick Kourie

16h00 - 16h30
"A Reusable Kernel for the Development of Control Software" W Fouche and P de Villiers, University of Stellenbosch

16h30 - 17h00
"An Implementation of Linda Tuple Space under the Helios Operating System" P G Clayton, E P Wentworth, G C Wells and F de-Heer-Menlah, Rhodes University

17h00 - 17h30
"The Design and Analysis of Distributed Virtual Memory Consistency Protocols in an Object Orientated Operating System" K MacGregor, University of Cape Town & R Campbell, University of Illinois at Urbana-Champaign

19h30 PRE-DINNER DRINKS

20h00 GALA CAPE DINNER
(Men: Jackets & ties)
WEDNESDAY 3rd July 1991

7h00 - 8h15 BREAKFAST

8h15 - 9h45 SESSION 3A

Venue: Hassner

Chairman: Assoc Prof P Wood

8h15 - 8h45
"Concurrency Control Mechanisms for Multidatabase Systems" A Deacon,
University of Stellenbosch

8h45 - 9h15
"Extending Local Recovery Techniques for Distributed Databases" H L Victor
& M H Rennhackkamp, University of Stellenbosch

9h15 - 9h45
"Analysing Routing Strategies in Sporadic Networks" S Melville,
University of Natal

9h45 - 10h15 TEA

10h15 - 11h00 SESSION 4

Venue: Hassner

Chairman: Prof P S Kritzinger
Invited paper: E Coffman

11h00 - 11h10 BREAK

SESSION 3B

Venue: Hassner C

Chairman: Prof G Finnie

8h15 - 8h45
"The Design of a Speech Synthesis System for Afrikaans" M J Wagener,
University of Port Elizabeth

8h45 - 9h15
"Expert Systems for Management Control: A Multiexpert Architecture"
V Ram, University of Natal

9h15 - 9h45
"Integrating Similarity-Based and Explanation-Based Learning"
G D Oosthuizen and C Avenant,
University of Pretoria
11h10 - 12h40 SESSION 5A

Venue: Hassner

Chairman: Prof C Bornman

11h10 - 11h40
"Efficient Evaluation of Regular Path Programs"
P Wood, University of Cape Town

11h40 - 12h10
"Object Orientation in Relational Databases"
M Rennhackkamp, University of Stellenbosch

12h10 - 12h40
"Building a secure database using self-protecting objects" M Olivier and S H von Solms, Rand Afrikaans University

SESSION 5B

Venue: Hassner C

Chairman: Prof A Krzesinski

11h10 - 11h40
"Modelling the Algebra of Weakest Preconditions"
C Brink & I Rewitsky, University of Cape Town

11h40 - 12h10
"A Model Checker for Transition Systems"
P de Villiers, University of Stellenbosch

12h10 - 12h40
"A New Algorithm for Finding an Upper Bound of the Genus of a Graph"
D I Carson and O R Oellermann, University of Natal

12h45-12h55 GENERAL MEETING of RESEARCH SYMPOSIUM ATTENDEES

Venue: Hassner

Chairman: Dr M H Linck

13h00 - 14h00 LUNCH

FINIS 6th COMPUTER SYMPOSIUM
PAPERS

of the

6TH RESEARCH SYMPOSIUM
PRODUCT FORM SOLUTIONS
FOR MULTISERVER CENTRES WITH
HIERARCHICAL CONCURRENCY CONSTRAINTS

AE Krzesinski
University of Stellenbosch
7600 Stellenbosch
South Africa

R Schassberger
Technische Universität Braunschweig
33 Braunschweig
Germany

July 1991

Abstract. The MultiServer centre with Hierarchical Concurrency Constraints (MSHCC) is an exponential multiserver with a queueing discipline which imposes restrictions on the number of customers of different types and subtypes that can be served simultaneously. The MSHCC centre is of interest because it represents a (not too exotic) quasi-reversible queue and can therefore be included in a product form network, and because it can be used to represent system features which could not previously be modelled using classical BCMP centres.


Keywords and phrases: blocking, concurrency constraints, product form solutions, queueing networks, queueing theory.

1 INTRODUCTION

This paper describes an extension of a queueing model first presented in [7]. Consider a service station serving customers of types $c$ where $c \in C$ and $C$ is a countable set. Customers of type $c$ arrive individually in independent Poisson streams with rate $\lambda_c$. The customers, whether waiting or in service, form a queue in the order of their arrival. Each customer presents a demand for service time distributed exponentially with mean $\mu$. The server gives service at certain positions of the queue. Upon entering service a customer is served to completion without interruption, whereupon the customer departs. When a service completion occurs, the corresponding customer departs, the gap in the queue is closed by the obvious shift, and the server scans the queue from the front searching for the first customer whose admission into service would not violate the following set of concurrency constraints. Firstly, maximally $B$ customers can be served simultaneously where $1 \leq B \leq \infty$. The queue can thus be viewed as having $B$ parallel servers. Next, the set of customer types $C$ is partitioned as $\{C_r; r \in R\}$ where $R$ is a countable
set and maximally $B_r$ customers whose types are in $C_r$ are allowed to be in service simultaneously where $r \in \mathcal{R}$ and $1 \leq B_r \leq \infty$. Finally, for each $r \in \mathcal{R}$ the set $C_r$ is partitioned as $\{C_{rs} : s \in S_r\}$ where $S_r$ is a countable set and maximally $B_{rs}$ customers whose types are in $C_{rs}$ are allowed to be in service simultaneously where $s \in S_r$ and $1 \leq B_{rs} \leq \infty$. The $C_{rs}$ can be further partitioned, with corresponding restrictions placed on the number of customers simultaneously in service, but we shall not go beyond the $C_{rs}$ as this is a straightforward generalization of what we shall do. The queue discipline can thus be described as FCFS subject to concurrency constraints. We have therefore named the queue the MultiServer centre with Hierarchical Concurrency Constraints (MSHCC).

Our main results are (i) a product form expression for the steady state distribution of the queue; (ii) the statement that the queue is quasi-reversible [5] and can therefore be included in a network of quasi-reversible centres; (iii) efficient recursive expressions for computing the performance measures of a MSHCC centre. The original model introduced in [7, 8] did not partition the $C_r$ and $B_r = 1$ for all $r$. In [2] and [3] this was extended to allow $B_r > 1$. The hierarchical partitioning imposed by the MSHCC server is thus new. Finally, the model employs a state description more detailed than that used previously in the references quoted above – which greatly benefits the analysis.

The paper is organized as follows. Section 2 defines a more general queueing model. The global balance equations for this model are presented and are decomposed into two sets of partial balance equations. The first set of partial balance equations has a product form solution. It is shown that this solution also satisfies the second set of partial balance equations if certain conditions are imposed on the queueing discipline. Section 3 presents the MSHCC concurrency constraints. It is shown that the MSHCC queue satisfies these conditions. The MSHCC centre is next shown to be quasi-reversible [5]. Therefore a queueing network consisting of BCMP [1] and MSHCC centres has a product form solution. Section 4 uses aggregation techniques to reduce the joint probability distribution to a computationally tractable form. Section 5 derives the performance measures for the MSHCC centre. Finally, a numerical example is presented which illustrates how the concurrency constraints affect the performance of customers competing for service at a MSHCC centre.

2 THE MODEL

Consider a queue with state space $S = \{0\} \cup_{n=1}^{\infty} C^n$ where $0$ denotes the empty queue and $C$ denotes a countable set of customer types. Let $(c_n, \ldots, c_1)$ denote the state of a queue of length $n$ where $c_i$ denotes the type of the customer in position $i$ in the queue where $1 \leq i \leq n$, $n \in \mathbb{N}$ and $c_i \in C$. For $1 \leq i \leq n$ define the functions

$$1_{i,n}(c_n \ldots c_1) = \begin{cases} 1 & \text{the customer in position } i \text{ is in service} \\ 0 & \text{the customer in position } i \text{ is not in service} \end{cases}$$
Define
\[ k(c_n \ldots c_1) = \sum_{i=1}^{n} l_{i,n}(c_n \ldots c_1) \geq 1 \]
so that \( k(c_n \ldots c_1) \) is the number of customers being served in \( (c_n \ldots c_1) \).

Let customers of type \( c \) arrive to the queue individually in independent Poisson streams with rate \( \lambda_c \) such that \( \sum_{c \in C} \lambda_c < \infty \). Let arriving customers join the tail of the queue and let the front of the queue be identified with position 1. Let service on each position served be non-interruptible and take place at the positive rate \( \mu_n \) where \( n \) is the length of the queue. When service terminates at position \( i \), let the ensuing state be \( (c_n \ldots c_{i+1} c_{i-1} \ldots c_1) \).

For the corresponding Markov process, which is obviously regular, the global balance equations are given by
\[
P(0) \sum_{c \in C} \lambda_c = \sum_{c \in C} P(c) \mu_1
\]
and
\[
P(c_n \ldots c_1) \left( \sum_{c \in C} \lambda_c + k(c_n \ldots c_1) \mu_n \right) = \lambda_{c_n} P(c_{n-1} \ldots c_1)
\]
\[
+ \sum_{c \in C} \sum_{i=0}^{n} P(c_n \ldots c_{i+1} c_c \ldots c_1) l_{i+1,n+1}(c_n \ldots c_{i+1} c_c \ldots c_1) \mu_{n+1}
\]
(1)
where the term \( \lambda_{c_n} P(c_{n-1} \ldots c_1) \) is replaced by \( \lambda_{c_1} \) if \( n = 1 \).

We wish to determine solutions for equation (1) which also satisfy the partial balance equations
\[
P(c_n \ldots c_1) k(c_n \ldots c_1) \mu_n = \lambda_{c_n} P(c_{n-1} \ldots c_1)
\]
(2)
Equation (2) immediately yields
\[
P(c_n \ldots c_1) = P(0) \prod_{i=1}^{n} \frac{\lambda_{c_i}}{\mu_i k(c_i \ldots c_1)}
\]
(3)
In order for equation (3) to satisfy equation (1) for any choice of the \( \lambda_c \) it is necessary and sufficient that equation (4) satisfies the partial balance equations
\[
P(c_n \ldots c_1) \lambda_c = \sum_{i=0}^{n} P(c_n \ldots c_{i+1} c_c \ldots c_1) l_{i+1,n+1}(c_n \ldots c_{i+1} c_c \ldots c_1) \mu_{n+1}
\]
(4)
or equivalently (by substituting equation (3) into equation (4)) that
\[
1 = \sum_{i=0}^{n} \frac{k(c_n \ldots c_1) \ldots k(c_{i+1} \ldots c_1)}{k(c_n \ldots c_{i+1} c_c \ldots c_1) \ldots k(c_c \ldots c_1)} l_{i+1,n+1}(c_n \ldots c_{i+1} c_c \ldots c_1)
\]
(5)
In order to find sufficient conditions for equation (5) to be satisfied, assume
Assumption 1 For all \((c_n \ldots c_1)\) and for all \(1 \leq i \leq n\)

\[ l_{i,n}(c_n \ldots c_1) = l_{i,i}(c_i \ldots c_1) \]

Assumption 1 thus implies that the composition of the queue up to position \(i\) determines whether the customer at position \(i\) is admitted into service. Applying assumption 1 to equation (5) yields

\[ 1 = \sum_{i=0}^{n} \frac{k(c_n \ldots c_1) \ldots k(c_{i+1} \ldots c_1)}{k(c_n \ldots c_{i+1}c_i \ldots c_1) \ldots k(c_{c_i} \ldots c_1)} l_{i+1,i+1}(cc_i \ldots c_1) \]  

(6)

Let the right hand side of equation (6) be denoted as \(V_c(c_n \ldots c_1)\) if \(n > 0\), and as \(V_c\) if \(n = 0\). Note that \(V_c = 1\). Next define

\[ V_c(c_n \ldots c_1) = \max(i : l_{i+1,n+1}(c_n \ldots c_{i+1}c_i \ldots c_1) = 1, 0 \leq i \leq n) \]  

(7)

and henceforth, for fixed \(c\) and fixed \((c_n \ldots c_1)\), write \(V_c(c_n \ldots c_1) = \nu\). Next assume that for all \(c\)

Assumption 2 If \(\nu < n\) then for all \(0 \leq i \leq \nu < j \leq n\)

\[ k(c_j \ldots c_{i+1}c_i \ldots c_1) = k(c_j \ldots c_1) \]

Assumption 3 If \(\nu = n\) then for all \(0 \leq i \leq n\)

\[ k(c_n \ldots c_{i+1}c_i \ldots c_1) = k(c_n \ldots c_1) + 1 \]

Assumptions 2 and 3 will later be shown to be satisfied when the system of hierarchical constraints described in section 3 is used to regulate the admission of customers into service at the MSHCC centre.

If \(\nu = n\) then application of assumption 3 yields

\[ V_c(c_n \ldots c_1) = \frac{1 + k(c_n \ldots c_1)V_c(c_n-1 \ldots c_1)}{1 + k(c_n \ldots c_1)} \]

and if \(\nu < n\) application of assumption 2 yields

\[ V_c(c_n \ldots c_1) = V_c(c_\nu \ldots c_1) \]

**THEOREM 2.1** Under the assumptions 1, 2 and 3 the Markov process describing the fluctuation of the queue has its global balance equations solved by equation (3), and equation (3) also satisfies the partial balance equations (2) and (4).
PROOF. Given $V_e = 1$ and assuming that it has already been shown that $V_e(c_k \ldots c_1) = 1$ for all $c$ and $(c_k \ldots c_1)$, then the proof of the theorem follows immediately by induction.

We now show that the queueing process is quasi-reversible by demonstrating that for all $c \in C$ (i) the arrival times of type $c$ customers form independent Poisson processes and (ii) the departure times of type $c$ customers form independent Poisson processes. Both of these conditions hold: (i) is part of the definition of the model and (ii) is a consequence of equation (4).

3 CONCURRENCY CONSTRAINTS

Consider the queue described in section 2. Let $x = (c_n \ldots c_1)$ denote the state of the queue and let $k(x), k_r(x)$ and $k_{rs}(x)$ denote the total number of customers, the total number of $C_r$ - and the total number of $C_{rs}$- customers served in state $x$ respectively. The MSHCC concurrency constraints require that

$$k(x) = \sum_{i=1}^{n} 1_{i,n}(c_n \ldots c_1) \leq B$$

$$k_r(x) = \sum_{i=1}^{n} 1_{i,n}(c_n \ldots c_1) 1(c_i \in C_r) \leq B_r$$

$$k_{rs}(x) = \sum_{i=1}^{n} 1_{i,n}(c_n \ldots c_1) 1(c_i \in C_{rs}) \leq B_{rs}$$

where $1(E)$ is the indicator function of the event $E$.

Assumption 4 The server, upon completing the service of a customer, searches the queue from the front (queue position 1) to the end for the first customer whose admission into service would not violate any of the MSHCC concurrency constraints.

THEOREM 3.1 Under the assumptions 1, 2, 3 and 4 the MSHCC queue is a special case of the queue described in section 2 and satisfies the conditions of theorem 2.1.

PROOF. The MSHCC queue is clearly a special case of the queue presented in section 2, with the functions $i_{i,n}(\cdot)$ all well defined in terms of the concurrency bounds $B, B_r$ and $B_{rs}$, and assumption 4 defining how the server searches the queue to select a customer to take into service. Given that arrivals join the end of the queue and that service is non-interruptible then assumption 4 clearly implies assumption 1. It remains to show that assumptions 2 and 3 hold.

As in theorem 2.1, for a fixed queue configuration $(c_n \ldots c_1)$, and for a fixed customer type $c$, let $\nu_c(c_n \ldots c_1) = \nu$. Clearly assumption 3 is satisfied. As for
assumption 2, observe that for $0 \leq i \leq \nu < j \leq n$

\[
k(c_j \ldots c_{\nu+1}cc \ldots c_1) = k(cc_{\nu} \ldots c_1) + \sum_{k=\nu}^{j} l_{k+1,j+1}(c_j \ldots c_{\nu+1}cc \ldots c_1)
\]

\[
= k(cc_{\nu} \ldots c_1) + \sum_{k=\nu}^{j} l_{k+1,j+1}(c_j \ldots c_{\nu+1}c_{\nu} \ldots c_1 + c_{\nu+1}c_{\nu} \ldots c_1)
\]

It is obvious that for $0 \leq i \leq \nu < n$

\[
k(c_{\nu} \ldots c_{i+1}cc \ldots c_1) = k(cc_{\nu} \ldots c_1)
\]

so that

\[
k(c_j \ldots c_{\nu+1}cc \ldots c_1) = k(c_j \ldots c_{i+1}cc \ldots c_1)
\]

It therefore remains to show that for $\nu < j \leq n$

\[
k(c_j \ldots c_{\nu+1}cc \ldots c_1) = k(c_j \ldots c_1)
\]

Let $x = (c_j \ldots c_{\nu+1}cc \ldots c_1)$ and label the customers in positions $\nu + 1$ and $\nu + 2$ as the tag-1 and tag-2 customers respectively. Let $x'$ result from $x$ by exchanging the tagged customers, and let $y$ result from $x'$ by removing the tag-1 customer and shifting forward to close the gap. Thus $y = k(c_j \ldots c_1)$. We intend to demonstrate that $k(x) = k(x') = k(y)$ which will prove equation (9) to be correct.

Assume that $c \in C_{rs}$. First consider the case where $c_{\nu+1} \in C_{rs}$. Then either $k(x_0) = B$ or $k_r(x_0) = B_r$ or $k_{rs}(x_0) = B_{rs}$ where $x_0 = (cc \ldots c_1)$ and “or” means “inclusive or”. The status served or waiting of the tagged customers changes as $x$ becomes $x'$ and the status of the untagged customers remains unaltered as $x$ becomes $x'$ becomes $y$. Thus $k(x) = k(x') = k(y)$ which proves equation (9).

Next consider the case where $c_{\nu+1} \notin C_{rt}$ for some $t \neq s$. Then either $k(x_0) = B$ or $k_r(x_0) = B_r$. Also $k_{rt}(x_0) < B_{rt}$ otherwise the tag-2 customer could not be served after $x$ becomes $x'$ and hence the tag-1 customer would be served after the exchange, contradicting the definition of $\nu$. Thus the status of the tagged customers changes as $x$ becomes $x'$ and the status of the untagged customers remains unaltered as $x$ becomes $x'$ becomes $y$. Thus $k(x) = k(x') = k(y)$ which proves equation (9).

Finally consider the case where $c_{\nu+1} \notin C_r$. Then $k(x_0) = B$. The tag-2 customer changes its status from $W$ to $S$ as $x$ becomes $x'$ otherwise the tag-1 customer would still be served after the exchange. Thus equation (9) follows by the previous argument which completes the theorem.

A closer look at the above theorem reveals an unforeseen fact, namely that the total number of customers served in the queue remains unchanged under a permutation of the customers in the queue. This is expressed in

**COROLLARY 3.2** For the MSHCC queue

\[
k(c_n \ldots c_1) = k(c_{\sigma(n)} \ldots c_{\sigma(1)})
\]

where $\sigma$ denotes any permutation of $(1 \ldots n)$ and $(c_n \ldots c_1)$ is any state of $S$. 

74
PROOF. It suffices to show that for $0 \leq 1 < n$ and for all $(c_n \ldots c_1)$
\[
k(c_n \ldots c_1) = k(c_{n-1} \ldots c_i + 1 c_i \ldots c_1)
\tag{11}
\] Let $\nu = \nu_{c_n}(c_{n-1} \ldots c_1)$ be defined as in section 2. If $\nu = n - 1$ then equation (11) follows directly from assumption 3. If $\nu = n - 1$ then equation (11) follows directly from assumption 2 for all $0 \leq i \leq \nu$.

For $i = \nu + 1$ write $x = (c_{n-1} \ldots c_\nu + 1 c_n c_\nu \ldots c_1)$ and $x' = (c_{n-1} \ldots c_\nu + 2 c_n c_\nu + 1 \ldots c_1)$. As shown in theorem 3.1, $k(x) = k(x')$ which proves equation (11) for $i = \nu + 1$. Starting from $x'$ exchange the customers in positions $\nu + 2$ and $\nu + 3$. This leaves the status (W or S) of all customers unchanged, which proves equation (11) for $i = \nu + 2$. Further such exchanges proves equation (11) for the remaining $i$.

4 AGGREGATION

The steady state distribution $P(c_n \ldots c_1)$ which presents the MSHCC centre in terms of the order of the customers is too detailed to be of practical use when computing the performance measures of the MSHCC centre. Equation (10) is the key for obtaining results concerning aggregated states. Define
\[
S_m = \{x \in S : m_c(x) = m_c, c \in C\}
\] where $m_c(x)$ is the number of type $c$ customers present when the queue is in state $x$, and where $\bar{m} = (m_c)_{c \in C}$ where $m_c \geq 0$. Let $\bar{m} - c$ be obtained from $\bar{m}$ by substituting $m_c - 1$ for $m$, this being defined only for $m_c > 0$.

Corollary 3.2 allows us to introduce an abuse of notation for the MSHCC queue, namely for $x \in S_m$ to write $k(\bar{m}) = k(x)$ to denote the total number of customers served in a state from $S_m$.

THEOREM 4.1 For the MSHCC queue
\[
k(\bar{m} + c) = \begin{cases} 
k(\bar{m}) & |\bar{m}| \geq B_r \\
k(\bar{m}) & |\bar{m}| < B_r \text{ and } m_{rs} \geq B_{rs} \\
k(\bar{m}) + 1 & |\bar{m}| < B_r \text{ and } m_{rs} < B_{rs}
\end{cases}
\tag{12}
\] where $c \in C_{rs}$.

PROOF. This is a direct consequence of the MSHCC service discipline. \qed

THEOREM 4.2 For the MSHCC queue
\[
k(\bar{m}) P(S_m) = \sum_{c \in C, m_c > 0} \rho_c P(S_{\bar{m} - c})
\tag{13}
\] where $\rho_c = \lambda_c / \mu |\bar{m}|$
5 MSHCC PERFORMANCE MEASURES

PROOF. The proof is given in [6].

Equation (13) permits the recursive calculation of $P(S_m)$. However, simpler recursions are available in the domain where $k(m) < B$. These recursions, which are presented in section 5, require the following lemma, where $m_r(x)$ and $m_s(x)$ denote the number of $C_r$- and $C_s$- customers present in state $x$, where $x \in S$.

**Lemma 4.1** For $x \in S$ such that $k(x) < B$

$$k_r(x) = \begin{cases} B_r & m_r(x) \geq B_r \\ \sum_{s \in S_r} m_s(x) \land B_r & m_r(x) < B_r \end{cases}$$

(14)

where $p \land q$ is the smaller of the two integers $p$ and $q$. Furthermore, if $k(x) < B$ and $m_r(x) < B_r$, then $k_s(x) = m_s(x) \land B_r$ for all $s \in S_r$.

**Proof.** This is a direct consequence of the MSHCC service discipline.

Lemma 4.1 implies that for $\bar{m}$ such that $k(\bar{m}) < B$ we can allow a further abuse of notation for the MSHCC queue, namely for $x \in S_{\bar{m}}$ to write $k_r(\bar{m}) = k_r(x)$ to denote the number of $C_r$- customers served in a state from $S_{\bar{m}}$. If in addition, $\bar{m}$ is such that $m_r(x) < B_r$ for $x \in S_{\bar{m}}$, then we can similarly write $k_{rs}(\bar{m}) = k_r(x)$ to denote the number of $C_{rs}$- customers served in a state from $S_{\bar{m}}$.

**Lemma 4.3** For $\bar{m}$ such that $k(\bar{m}) < B$

$$k_r(\bar{m})P(S_{\bar{m}}) = \sum_{c \in C_r, m_c > 0} \rho_c P(S_{\bar{m}-c})$$

(15)

for all $r$. Additionally, if $m_r(x) < B_r$ for $x \in S_{\bar{m}}$, then for this $r$ and for all $s \in S_r$

$$k_{rs}(\bar{m})P(S_{\bar{m}}) = \sum_{c \in C_r, m_c > 0} \rho_c P(S_{\bar{m}-c})$$

(16)

**Proof.** The proof is given in [6].

5 MSHCC PERFORMANCE MEASURES

This section presents efficient recursions to compute the performance measures at a MSHCC centre. The recursions employ equation (12) and are confined to the domain $1 \leq k(\bar{m}) < B$ so that equations (14) and (15) can be exploited. The definitions and theorems in this section therefore all assume that $1 \leq k(\bar{m}) < B$.

Recall $\bar{m} = (m_c)_{c \in C}$ and define $\bar{m}_c = (m_c)_{c \in C}$, where $m_c \geq 0$. Define $|\bar{m}| = n$. Let $\bar{m}_r - c$ be obtained from $\bar{m}_r$ by substituting $m_c - 1$ for $m$, this being defined
only for \( m_c > 0 \). Let \( M_c \) denote a random variable distributed as the number of type c customers present in the queue at equilibrium. Define a random variable \( \bar{M}_r = (M_c)_{c \in C_r} \). Define

\[
P_b^{m_r}(r + 1, n) = \Pr(k(\bar{m}) = b, \bar{M}_r = \bar{m}_r, |\bar{M}_i| = 0 \text{ for } i > r)
\]

Define \( \bar{B}_r = (B_{rs})_{s \in S_r} \) and denote \( \bar{m}_r < \bar{B}_r \) if \( m_{rs} < B_{rs} \) for all \( s \in S_r \) and \( \bar{m}_r \geq \bar{B}_r \) otherwise. Let \( \rho_c = \lambda_c / \mu_c \).

**THEOREM 5.1** For \( 1 \leq r < R \) and \( \bar{m}_r < \bar{B}_r \)

\[
|\bar{m}_r| P_b^{m_r}(r + 1, n) = \sum_{c \in C_r, m_c > 0} \rho_c P_b^{m_r-c}(r + 1, n - 1)
\]

**THEOREM 5.2** For \( 1 \leq r < R \) and \( \bar{m}_r \geq \bar{B}_r \) and \( |\bar{m}_r| < B_r \)

\[
k_s(\bar{m}) P_b^{m_r}(r + 1, n) = \sum_{s \in S_r} \sum_{m_{rs} = B_{rs}} \rho_c P_b^{m_r}(r + 1, n - 1)
\]

\[+ \sum_{s \in S_r} \sum_{m_{rs} = B_{rs}} \rho_c P_b^{m_r-c}(r + 1, n - 1)
\]

\[+ \sum_{s \in S_r} \sum_{m_{rs} > B_{rs}} \rho_c P_b^{m_r-c}(r + 1, n - 1)
\]

**THEOREM 5.3** For \( 1 \leq r < R \) and \( |\bar{m}_r| \geq B_r \)

\[
B_r P_b^{m_r}(r + 1, n) = \sum_{c \in C_r, m_c > 0} \rho_c P_b^{m_r-c}(r + 1, n - 1)
\]

\[+ \sum_{s \in S_r} \sum_{m_{rs} = B_{rs}} \rho_c P_b^{m_r-c}(r + 1, n - 1)
\]

\[+ \sum_{s \in S_r} \sum_{m_{rs} > B_{rs}} \rho_c P_b^{m_r-c}(r + 1, n - 1)
\]

For \( 1 \leq r < R \) define

\[
P_b(r + 1, n) = \Pr(k(\bar{m}) = b, |\bar{M}_i| = 0 \text{ for } i > r)
\]

Clearly

\[
P_b(r + 1, n) = P_b(r, n) + \sum_{m_r |\bar{m}_r| > 0} P_b^{m_r}(r + 1, n)
\]
where the $P^m_b(r+1, n)$ are given in theorems 5.1, 5.2 and 5.3 above. These theorems thus yield a recursive procedure to compute $P_b(R, n)$.

Next define

$$P^m_R(n) = \Pr(k(m) = b, M_R = m_R)$$

**THEOREM 5.4** For $m_R < B_R$

$$|m_R|P^m_R(n) = \sum_{c \in C_R, m_c > 0} \rho_c P^m_{b-1} - c(n - 1)$$

**THEOREM 5.5** For $m_R \geq B_R$ and $|m_R| < B_R$

$$k_R(m)P^m_R(n) = \sum_{s \in S_R} \sum_{c \in C_R, m_c > 0} \rho_c P^m_{b-1} - c(n - 1)$$

**THEOREM 5.6** For $|m_R| \geq B_R$

$$B_R P^m_R(n) = \sum_{c \in C_R, m_c > 0} \rho_c P^m_{b-1} - c(n - 1)$$

Next define

$$P_b(n) = \Pr(k(m) = b)$$

Clearly

$$P_b(n) = P_b(R, n) + \sum_{m_R, |m_R| > 0} P^m_R(n)$$
where the $P_{b}^{m_{R}}(n)$ are given in theorems 5.4, 5.5 and 5.6 above and $P_{b}(R,n)$ is given in equation (17). We thus have a recursive procedure to compute $P_{b}(n)$ for $1 \leq b < B$.

Let $L_{b}(n)$ denote the average number of type $C_{R}$ customers at the MSHCC centre when $k(\bar{m}) = b$. Then for $1 \leq b < B$

$$L_{b}(n) = \sum_{m_{R}} |m_{R}| P_{b}^{m_{R}}(n)$$

Let $L(n)$ denote the average number of type $C_{R}$ customers at the MSHCC centre.

**THEOREM 5.7**

$$BL(n) = \sum_{b=1}^{B-1} (B - b)L_{b}(n) + \rho L(n - 1) + \rho R$$

where $\rho R = \sum_{c \in C} \rho_{c}$ and $\rho = \sum_{c \in C} \rho_{c}$.

The proofs of theorems 5.1 through 5.7 are given in [3].

### 6 APPLICATIONS

**MULTIPORTED MEMORY**

Consider [7, 8] a computer system consisting of $N$ processors accessing $K$ memory modules via a partitioned multiple bus system. Each of the $G$ groups of $B$ buses gives access to a subset of $K/G$ memory modules. Each memory module $k$ is $n_{k}$-ported so that maximally $n_{k}$ processors can access memory module $k$ simultaneously. The system is modelled as a queuing network consisting of an IS centre representing the processors and $G$ MSHCC centres, each representing one group of buses and the associated memory modules. Each MSHCC centre consists of $B$ servers which represent the $B$ buses in its group. The customers in the network belong to $K$ classes.

A processor service interval followed by a data transfer to/from memory module $k$ is modelled as a customer departing from the processor service centre, and moving to the $g^{th}$ MSHCC centre where group $g$ contains an access path to memory module $k$. The customer changes class to class $k$ and queues for service at the MSHCC centre. The class $k$ customer enters into service if one of the $B$ servers is free (a bus is available) and if at most $n_{k}$ class $k$ customers are in service (at most $n_{k}$ processors can access memory module $k$ simultaneously). The fact that different classes of customers are permitted allows each processor to preferentially access a different set of memory modules and to issue memory access requests at a rate differing from the request rates of the other processors.
A MESSAGING CARD

In some distributed architectures such as telephone switching exchanges, the messaging function between a high level peripheral (decentralized call processor) and a lower level one (line or trunk controller) is performed on the high level side by a specialized processor (the messaging card) which controls simplex channels to the lower level peripherals. The processor time is partitioned into \( B \) fixed time slots, each of which is allocated to a process whose function is to send outgoing messages. The exchange forwards messages to \( K \) destinations, each of which is reachable on any of \( n_k \) channels. When a buffer is queued for transmission it has to wait for a process to be available to service the request and for an outgoing channel to be free. The messaging card can be modelled by a MSHCC centre consisting of \( B \) servers (the transmission processes) serving customers belonging to \( K \) classes, each with its own concurrency limit \( n_k \).

MULTILAYER WINDOW FLOW CONTROL

In X.25 packet switched networks window flow control is implemented at two levels. At the data link level maximally \( B \) frames may be unacknowledged between two communicating nodes; at the network level maximally \( n_k \) packets may be unacknowledged on each of \( K \) virtual circuits multiplexed onto the data link.

A MSHCC centre can be used to model the above window flow control policy as follows [4]. The customers arriving at the MSHCC centre belong to \( K \) groups. A customer of group \( k \) represents a packet belonging to virtual circuit \( k \). Packets are queued for transmission in the order of their arrival. A customer of group \( k \) will be admitted into service (transmitted on the link) provided that fewer than \( B \) servers are occupied and maximally \( n_k - 1 \) other group \( k \) customers are in service. A customer occupies a server for a time corresponding to the transmission delay for a packet and the time taken for an acknowledgement to be returned to the node. Thus maximally \( B \) packets may be in transit at the data link level and maximally \( n_k \) at the network level for each virtual circuit \( k \).

MULTISERVER MULTQUEUE SYSTEMS

A Multiserver MultiQueue system (MSMQ) consists of a set of \( N \) queues and \( S \) servers. Customers arrive to the queues individually in independent Poisson streams. The queues have infinite capacities and the queuing discipline within each queue is FCFS. Servers move cyclically from one queue to another and every time a server visits a queue at most one waiting customer is served. The movement of servers from queue to queue takes place in zero time. Two server utilization policies are possible: the \( 1 \times Q \) policy states that only 1 server can be serving a queue at a given time and the \( S \times Q \) policy permits all \( S \) servers to simultaneously attend to one queue.
The dynamics of an MSMQ system operating according to the $1 \times Q$ policy is remarkably similar to the MSHCC queue with $S$ servers and $N$ customer types. The customer types can be identified with the MSMQ queues. In each case maximally one customer per type or queue can be in service simultaneously. The only difference is in the polling order: in the case of MSHCC it is FCFS subject to the concurrency constraints, and in the MSMQ system it is cyclical. However, if the scheduling algorithm minimizes the number of empty queues, the impact of the polling order on the average customer delay can be expected to be minimal.

The similarity between the MSHCC and MSMQ systems suggests that the MSHCC queue can provide a very accurate approximation of the $1 \times Q$ MSMQ system. Recent simulation studies [9] confirm that the accuracy of the MSHCC approximation to the $1 \times Q$ system usually lies within the confidence interval of the simulation process.

References


