DESUPER DEORSUM

CELERIORE EXISTIMATE ET RECTIORE FACTO

Quaestiones Informaticae

Vol. 2 No. 3 September, 1983

Quaestiones Informaticae

An official publication of the Computer Society of South Africa and of the South African Institute of Computer Scientists

'n Amptelike tydskrif van die Rekenaarvereniging van Suid-Afrika en van die Suid-Afrikaanse Instituut van Rekenaarwetenskaplikes

Editor: Prof G. Wiechers
Department of Computer Science and Information Systems
University of South Africa
P.O. Box 392, Pretoria 0001

Subscriptions

Annual subscriptions are as follows:

	SA	US	UK
Individuals	R6	$7	£3,0
Institutions	R12	$14	£6,0

Circulation manager
Mr E Anderssen
Department of Computer Science
Rand Afrikaanse Universiteit
P O Box 524
Johannesburg 2000
Tel.: (011) 726-5000

Quaestiones Informaticae is prepared for publication by Thomson Publications South Africa (Pty) Ltd for the Computer Society of South Africa and the South African Institute of Computer Scientists.

An Analytical Model of a Mixed-Workload MVS Computer System

R.J. Mann
Comcon (Pty) Limited, Johannesburg

Abstract
A basic description of the facilities of the computer modelling package, AUTO-CONFIGURATER, is presented together with detailed procedures to calculate the required input parameters from monitor statistics. Techniques to calibrate the model are described together with an investigation into how the modelling package enables configuration and tuning alternatives to be evaluated.

1. The need for capacity planning

New computer applications are being developed in greater numbers than ever before. In addition, the workload growth for existing systems is constantly expanding.

In order to provide adequate hardware and software to accommodate demand, the capacity planner has to forecast performance bottlenecks and capacity shortfalls and evaluate alternative scenarios to address capacity problems by investigating various configuration and workload possibilities, using "what if" questions. In this way the planner can predict such performance indicators as response time, system throughput rates and device utilisations.

The AUTO-CONFIGURATER system (AUT83) described here provides a cheap and easy-to-use tool for these activities.

2. Auto-Configurater Theoretical background

Queueing network models have been successfully used for modelling computer systems and computer communications systems for some years, (eg. BUZ73).

The Mean Value Analysis (MVA) algorithm of Reiser and Lavenberg (REI78) avoids the problem of overflow of earlier techniques and presents an intuitively simple interpretation of flow through the system. It requires an excessive amount of main memory for all but the smallest models.

AUTO-CONFIGURATER uses A CONVERGENCE algorithm together with heuristics to handle a number of specific conditions found in computer systems including paging, blocking, class priorities, missed disk rotations and disk/channel overlap. (WUL82) This algorithm requires very little memory for solution.

3. Creating the MVS System Profile

3.1 Defining the time period

The analyst must inspect all periods of operation of the computer system to determine the appropriate interval. In macro terms he must evaluate the annual calendar to identify year-end, quarterly or monthly peaks. Thereafter, he must analyse each 24 hour period to determine basic modes of operation. For example, the night shift may be exclusively batch processing whereas the day shift consists of online processing. Based on this analysis of workload types and peaks, the analyst must select the appropriate time period which the model must cover which will generally be the peak period of the shift.

3.2 Sources of statistics

The basic starting lines used to build the model are derived from RMF. These are supplemented by SMF and IMS (or other DB/DC) statistics.

3.3 Defining the auto configurater model

The model should be defined with well in excess of the actual numbers of servers, classes, disk types and limits, (to allow further expansion of the basic system profile).

The following subsections define elements in building a model. The model topology is presented in FIGURE 1 and model dialogue is presented in FIGURE 2.

3.3.1 Model class definitions

The workload which runs in the machine can be classified into a number of different classes. The basic criteria which should be used for class definition are the use of common software or transactions of common duration. Examples are TSO, Batch, IMS short transactions and IMS long transactions.

It should always be borne in mind that the effect involved in model building increases with the number of classes defined. The analyst must evaluate the trade-off of increased accuracy versus increased effort in defining the model.

Each defined class is assigned a numeric priority value. (The higher the priority, the larger the priority value assigned). These class priorities are similar to operating system priorities assigned to jobs of these classes and we used *classes* in admitting tasks to the computer system or in scheduling the CPU.

3.3.2 Defining limits

The LIMIT command defines how tasks of the various classes are to be admitted into the computer system. Each limit command defines a limit on the number of tasks overall which may be admitted for the specified classes.

3.3.3 Defining disk types

The DISK command is entered for each disk type in the system (eg. 3350 or 3380). Default seek, rotation and transfer times are specified in each command. Unless these values are overridden for any particular disk server, they are used unchanged for that server. Supplier averages should be used.

3.3.4 Disk service times

The RMF report lists average service time (AST) for each disk drive.
Note that:

AST = seek + latency + missed rotations + transfer
latency = R/2 where R = rotation

$$\text{missed rotations} = \frac{UR}{1\text{-}U}$$

where U = utilisation of channel
and transfer = the default for the device type

3.3.5 Disk channels

Each logical channel is modelled as a channel using mean service time and utilisations of all physical channels which comprise the logical channel.

3.3.6 Calculation of class-dependent IOs and CPU service time

CPU service time and server IOs are specified by class. This means that total CPU seconds and total IOs at each device must be apportioned to each class. Ideally, this should be, as they are actually incurred during the processing of tasks of each class. In practice, this can never be done exactly. The degree of accuracy of these parameters is dependent on the amount of effort by the analyst to calculate them.

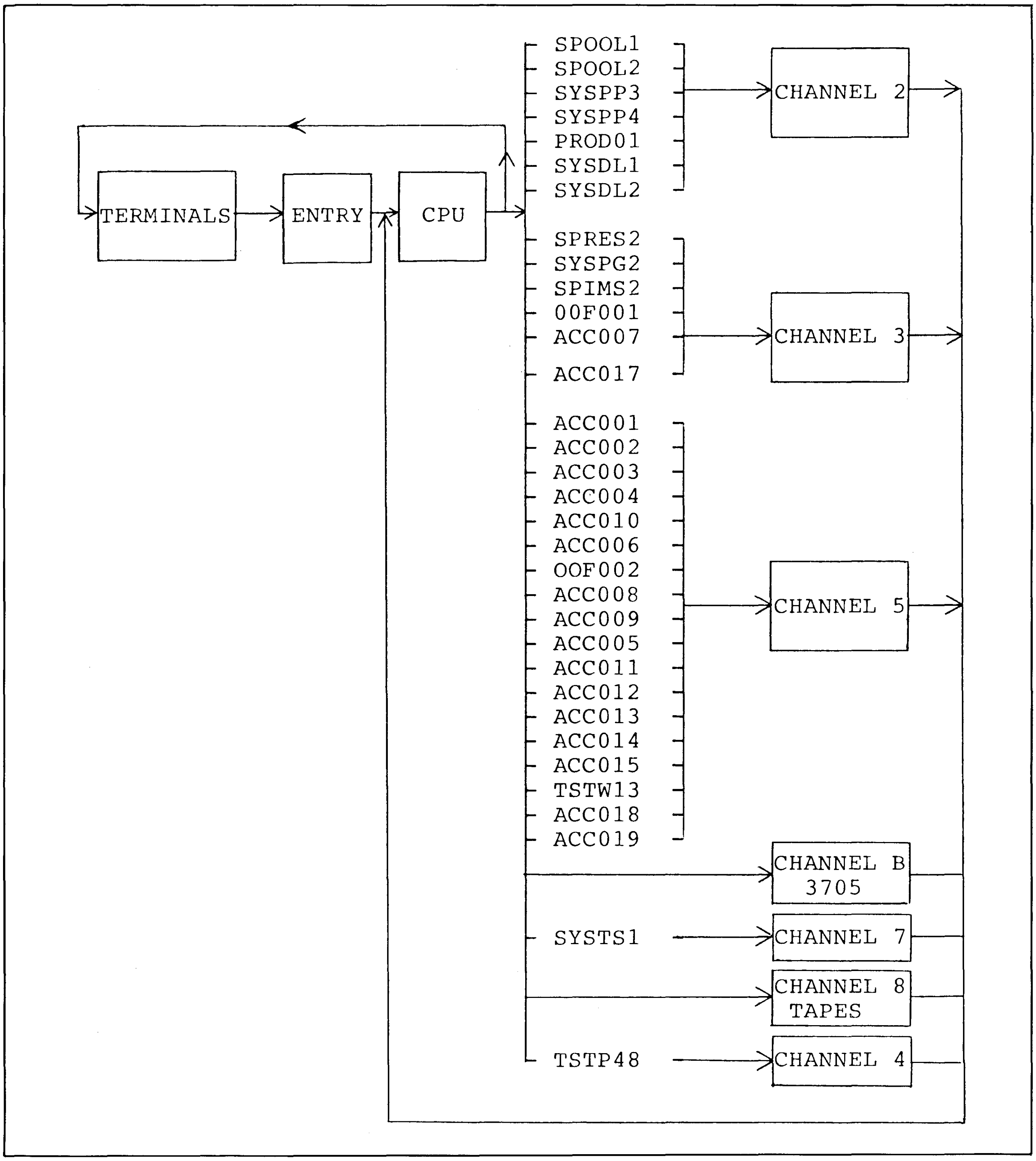

FIGURE 1 — QNA model topology.

A. The Detailed Method Using Accounting Statistics

Accounting statistics are machine statistics showing resource consumption by job or transaction. They include SMF statistics as well as DB/DC statistics. Usually, the SMF statistics (CPU seconds and IOs per device) are not complete since many operating system attributable components are not recorded by SMF.

The total CPU seconds and IOs recorded by SMF will be less than the consumption recorded by RMF. IBM has defined capture ratios for various workload types to apply to the SMF statistics in order to bring them into line with RMF data.

The SMF totals are used as the basis for allocating the more accurate RMF statistics.

RMF provides total CPU seconds used and IOs by device in the sampling period. The analyst must use the ratio of the usage for a particular class to the total usage as recorded by SMF as the basis for allocating RMF data.

B. An Approximate Technique Using RMF Only

The RMF Performance Group report contains summaries of IO, CPU and SRB usage by performance group. We assume using this technique that each class consists of one or more performance group and that no performance group is used by two or more classes.

Service definition coefficients for each of CPU, SRB and IOC (IOC refers to device IOs and CPU and SRB together constitute CPU service) are defined on each Performance Group report at the top of each page. These are used to calculate total CPU

```
NEW=BASE01,NOCL=40,NSER=100,NLIM=10,NDSK=3                             -(1)
MEMORY=2862,TIME=28800,A=4.585,K=2,MIPS=4200000                        -(2)
DISK=3330,SEEK=0,03,ROT=0.0167,TFR=0.004                               -(3)
DISK=3350,SEEK=0.025,ROT=0,0167,TFR=0.004
CLASS=1,NAME=ONLINE,NO=0.99,PR=2                                       -(4)
CLASS=2,NAME=BATCH,NO=0.07,PR=1
CLASS=3,NAME=ENVIRO,NO=1.2,PR=3
LIMIT=1,VAL=19                                                         -(5)
LIMIT=2,VAL=10,SI2=NO,SI3=NO
LIMIT=3,VAL=3,SI1=NO,SI3=NO
SERVER=TERMINAL,COST=754,LOC=OUT,IO1=492000,IO2=1,IO3=1                -(6)
SERVER=CPU,COST=253,ST1=0.0046,ST2=0.0042,ST3=0.0034
SERVER=CHAN2,COST=0.4,S=0.002
SERVER=CHAN3,COST=0.4,S=0.005
SERVER=CHAN4,COST=0.4,S=0.010
SERVER=CHAN5,COST=0.4,S=0.002
SERVER=CHAN7,COST=0.4,S=0.020
SERVER=CHAN8,COST=0.4,S=0.001,IO1=68134,IO2=5925,IO3=74059
SERVER=CHANB,COST=.4,S=.001,IO1=351216
SERVER=SPOOL1,COST=2.2,CHAN=CHAN2,DISK=3350,IO1=10457,IO2=742,
  IO3=11425,
SEEK=0.005
SERVER=SPOOL2,COST=2.2,CHAN=CHAN2,DISK=3350,IO1=514,IO2=36,IO3=562,
SEEK=0.0001
SERVER=SYSPP5,COST=2.2,CHAN=CHAN2,DISK=3350,IO1=618,IO2=44
  IO3=676,SEEK=0.048
PAGE,DEV=SPRES2,DEV=SYSPG2,DEV=SPIMS2                                  -(7)
SPLIT=SPRES2,FRAC=0.531                                                -(8)
SPLIT=SYSPG2,FRAC=0.227
SPLIT=SPIMS2,FRAC=0.242
SOLVE                                                                  -(9)
STATISTICS=TEST,TYPE=ALL                                               -(10)
SERVER=PROD01,IO1=72616,IO2=5153,IO3=79340                             -(11)
SERVER=OOF001,IO1=74459,IO2=5284,IO3=81354
SOLVE                                                                  -(12)
STATISTICS=REMOVE DISKLOAD,TYPE=ALL                                    -(13)
SAVE=MODEL                                                             -(14)
END
```

– (1)	define new model	– (6)	define servers	– (11)	change server data
– (2)	define memory available	– (7)	define paging devices	– (12)	solve model
– (3)	define disk types	– (8)	define allocation of paging to devices	– (13)	print all statistics
– (4)	define classes	– (9)	solve model	– (14)	save model on library
– (5)	define blocking limits	– (10)	print all statistics		

FIGURE 2 — Sample Model Dialogue.

and IO usage ratios by class. The simplifying assumption is then made that all classes will use all devices in the same ratio as calculated here. These ratios are applied to the device IO totals and total CPU seconds presented by RMF.

3.3.7 Paging

The fraction of paging to any one page device is the total paging IOs to that device divided by the total paging IOs to all devices. The SPLIT command is used to define these calculated fractions.

3.3.8 Number of tasks per class

The basic principle used is that the number of tasks in each class must be adjusted until the total number of transactions of the model approximate those measured in the statistics.

Let M_i = no of tasks in class i
E = total elapsed time
R_i = response time of class i
N_i = total number of transactions for class i actually measured

Then for all classes:

$M_i = N_i R_i / E$

The model assumes a starting value for R_i and then automatically iterates until convergent values of M_i are found.

4. Calibration and Prediction Modelling

We have compared measured performance values with the performance values calculated by the model. Using the principle established by (BUZ78) that if reasonable agreement is observed under a series of conditions the model can be considered capable of predicting system performance, under related alternative conditions.

Our results are presented in FIGURE 3. As a presentation method we use the cost utilisation histogram (BOR77) together with three performance measures, F, B and P.

The F value is defined to be the resource utilisation indicator. The value B is defined as a cost-imbalance factor which ranges from 0 for a perfectly balanced system to 1 for a system totally out of balance and the P factor is a measure of system balance unweighted by cost and also varies between 0 and 1.

By examination of the device utilisations the bottleneck device was identified to be disk PROD01. We can eliminate this bottleneck by splitting a reasonable proportion of the disk I/O's to a low utilised device, in this case disk 00F001. We also modelled the effects of a CPU upgrade, moving from an IBM 3033 to a 3081XA. A summary of results is given in FIGURE 4 which shows both individual and cumulative effects on performance measures.

Utilisations

Device	RMF Actual	Model	% Error
CPU	.267	.272	1,87
CHAN2	.028	.027	3,57
CHAN3	.023	.024	4,35
CHAN4	.0	.0	—
CHAN5	.072	.076	5,56
CHAN7	.0	.0	—
CHAN8	.001	.006	*
CHANB	.0	.008	–
SPOOL1	.017	.016	5,88
SPOOL2	.001	.001	0
SYSPP3	.0	.001	–
SYSPP4	.0	.0	—
SYSPP5	.0	.003	–
PROD01	.349	.349	0
SYSDL1	.0	.0	—
SYSDL2	.0	.0	—
SPRES2	.001	.005	*
SYSPG2	.001	.003	*
SPIMS2	.0	.001	—
OOF001	.0	.002	—
ACC007	.082	.081	1,22
ACC017	.013	.011	15,39
TSTP48	.0	.0	—
ACC001	.047	.051	8,51
ACC002	.063	.082	30,16
ACC003	.0	.001	—
ACC004	.109	.122	11,93
ACC010	.036	.039	8,33
ACC006	.094	.104	10,64
OOF002	.0	.002	—
ACC008	.045	.049	8,89
ACC009	.038	.041	7,89
ACC005	.105	.118	12,38
ACC011	.024	.027	12,50
ACC012	.105	.116	10,48
ACC013	.105	.118	12,38
ACC014	.027	.029	7,41
ACC015	.011	.012	9,09
TSTW13	.0	.0	—
ACC018	.046	.051	10,87
ACC019	.103	.116	12,62
SYSTS1	.0	.0	—
PAGE RATE/ SEC	.28	.271	3,21

*High Relative Error in cases of low utilisation due to rounding error.

FIGURE 3 — Model calibration.

	Utilisations CPU	PROD01	00F001	System Throughput	% Improvement
Base Model	0.27	0.349	0.002	0.0000344	—
Scenario 1	0.29	0.185	0.099	0.0000369	7,27
Scenario 2	0.16	0.375	0.002	0.0000361	4,94
Scenario 1 and 2	0.17	0.201	0.108	0.0000383	11,34

FIGURE 4 — A summary of results

References

[1] AUT83 AUTO-CONFIGURATER Reference Manual, Comcon International, 1983.

[2] BOR77 I. Borovits and P. Ein-Dor, Cost/Utilisation: A Measure Of System Performance. Comm. ACM 20, No. 3, 1977.

[3] BUZ73 J. P. Buzen, Computational Algorithm for Closed Queueing Networks with Operational Servers. Comm. ACM 16, No. 9, 1973.

[4] BUZ78 J. P. Buzen, A Queueing Network Model of MVS. ACM Computing Surveys 10, No. 3, 1978.

[5] REI78 M. Reiser and S. S. Lavenberg, Mean Value Analysis of User Multichain Queueing Networks. IBM Research Report RC7023, 1978.

[6] WUL82 S. Wulf, Performance Prediction During Database System Development. Ph.D. Thesis, University of South Africa, 1982.

Notes for Contributors

The purpose of this Journal will be to publish original papers in any field of computing. Papers submitted may be research articles, review articles, exploratory articles of general interest to readers of the Journal. The preferred languages of the Journal will be the congress languages of IFIP although papers in other languages will not be precluded.

Manuscripts should be submitted in triplicate to: Prof. G. Wiechers at:

Department of Computer Science
University of South Africa
P.O. Box 392
Pretoria 0001
South Africa

Form of manuscript

Manuscripts should be in double-space typing on one side only of sheets of A4 size with wide margins. The original ribbon copy of the typed manuscript should be submitted. Authors should write concisely.

The first page should include the article title (which should be brief), the author's name, and the affiliation and address. Each paper must be accompanied by a summary of less than 200 words which will be printed immediately below the title at the beginning of the paper, together with an appropriate key word list and a list of relevant Computing Review categories.

Tables and figures

Illustrations and tables should not be included in the text, although the author should indicate the desired location of each in the printed text. Tables should be typed on separate sheets and should be numbered consecutively and titled.

Illustrations should also be supplied on separate sheets, and each should be clearly identified on the back in pencil with the Author's name and figure number. Original line drawings (not photoprints) should be submitted and should include all relevant details. Drawings, etc., should be submitted and should include all relevant details. Drawings, etc., should be about twice the final size required and lettering must be clear and "open" and sufficiently large to permit the necessary reduction of size in block-making.

Where photographs are submitted, glossy bromide prints are required. If words or numbers are to appear on a photograph, two prints should be sent, the lettering being clearly indicated on one print only. Computer programs or output should be given on clear original printouts and preferably not on lined paper so that they can be reproduced photographically.

Figure legends should be typed on a separate sheet and placed at the end of the manuscript.

Symbols

Mathematical and other symbols may be either handwritten or typewritten. Greek letters and unusual symbols should be identified in the margin. Distinction should be made between capital and lower case letters between the letter O and zero; between the letter l, the number one and prime; between K and kappa.

References

References should be listed at the end of the manuscript in alphabetical order of author's name, and cited in the text by number in square brackets. Journal references should be arranged thus:

1. ASHCROFT, E. and MANNA, Z. (1972). The Translation of 'GOTO' Programs to 'WHILE' Programs, in *Proceedings of IFIP Congress 71,* North-Holland, Amsterdam, 250-255.
2. BÖHM, C. and JACOPINI, G. (1966). Flow Diagrams, Turing Machines and Languages with only Two Formation Rules, *Comm. ACM,* 9, 366-371.
3. GINSBURG, S. (1966). *Mathematical Theory of context-free Languages,* McGraw Hill, New York.

Proofs and reprints

Galley proofs will be sent to the author to ensure that the papers have been correctly set up in type and not for the addition of new material or amendment of texts. Excessive alterations may have to be disallowed or the cost charged against the author. Corrected galley proofs, together with the original typescript, must be returned to the editor within three days to minimize the risk of the author's contribution having to be held over to a later issue.

Fifty reprints of each article will be supplied free of charge. Additional copies may be purchased on a reprint order form which will accompany the proofs.

Only original papers will be accepted, and copyright in published papers will be vested in the publisher.

Letters

A section of "Letters to the Editor" (each limited to about 500 words) will provide a forum for discussion of recent problems.

Hierdie notas is ook in Afrikaans verkrygbaar.

Quaestiones Informaticae

Contents/Inhoud

*Presented at the second South African Computer Symposium held on 28th and 29th October, 1981.