

Chapter 17

Constructing a Model from RMF Data

17.1. Introduction

In Chapter 12 we described in general terms how to determine the input parameter values of a queueing network model from knowledge of a system, measurement data, and accounting data. In this chapter we are more specific: we consider the determination of parameter values for models of computer systems running IBM's MVS operating system, using information obtained from the MVS Resource Measurement Facility (RMF). We choose MVS for special treatment for several reasons:

- More large installations run MVS than any other single operating system.
- Many performance analysts who work with other systems have MVS experience in their backgrounds.
- The measurement and monitoring facilities associated with MVS have greater variety and sophistication than those of most other systems.

While this chapter will be of greatest utility to those involved with MVS and RMF, the techniques and difficulties that we illustrate are similar to those that arise in the context of many systems. To facilitate understanding by persons not familiar with MVS and RMF, Sections 17.2 and 17.3 provide introductions to the concepts and terminology associated with them. Of necessity, our discussions are relatively superficial, and pertain to specific releases of MVS and RMF.

Following the structure of Chapter 12, Sections 17.4, 17.5, and 17.6 treat customer description, center description, and service demands, respectively. Section 17.7 indicates how performance measures can be derived from RMF reports for the purpose of model validation. In each of these sections we first describe the techniques used to determine the corresponding parameter values from RMF reports, and then, in a "double-boxed" paragraph, illustrate these techniques in the context of a specific example. This example is based on standard RMF reports from an installation running MVS on an Amdahl 470 V/8 with 16 megabytes of main memory, 12 physical channels, and roughly 150 IBM 3350 disk

drives. The workload consists of two components: interactive (TSO) and batch. The RMF reporting interval was one hour during an afternoon peak load period. This system is much simpler than many large MVS installations (for example, it has fewer workload components). It suffices, however, to illustrate the basic parameter determination techniques.

As we have emphasized throughout the book, the goals of a particular modelling study must be taken into account in the construction of a model. In our example we assume that the system changes to be investigated are moderate changes to workload intensities. There are two significant implications of this assumption:

- We need not include a sophisticated representation of the disk I/O subsystem, as described in Chapter 10, since we would not expect the various components of disk service demands to change significantly under the modifications being investigated.
- We do not require a careful breakdown of paging activity by workload component (difficult to obtain from RMF alone), since we would not expect the level of paging activity to change significantly.

17.2. Overview of MVS

In this section we introduce some aspects of IBM's MVS operating system, which runs in many major computer installations. MVS has several components and features that relate to performance, and thus are important to the modelling of MVS systems.

Workload components in MVS are defined in the *Installation Performance Specification (IPS)*. A set of *performance groups* is established, each of which optionally is divided into a set of *performance periods*. An incoming transaction, based on its identity, enters the first performance period of some particular performance group.

A *service objective* is associated with each performance period, which states the desired rate at which *service units* are acquired by transactions belonging to that period. Service units are computed as the weighted sum of logical I/O operations, main storage occupancy (in units of 50 kilobyte-seconds), and CPU service (measured in 100ths of a second and adjusted by a factor reflecting the speed of the processor). The weights, called *service definition coefficients*, are set by the installation manager.

The *System Resources Manager (SRM)* controls the allocation of resources. The SRM's decisions are based on the progress of transactions relative to their associated service objectives. As transactions reach specified thresholds of attained service, they move from one performance period to the next. The service objectives of successive performance

periods call for lower and lower priority for resource allocation. The thresholds are chosen so that most of the transactions that enter a period complete within that period. This has the effect of providing good response times to short transactions by discriminating against longer transactions.

Competition for memory is handled by defining a set of *domains*, establishing a limit on the number of transactions that can be active simultaneously in each domain (the *domain capacity*), and associating each performance period with some domain. Transactions in periods associated with the same domain compete with one another for memory.

Typical performance groups defined in MVS installations include batch (possibly split into components such as production and test) and TSO. Other performance groups correspond to *started tasks* (jobs that never terminate). Started tasks may include such major subsystems as the IMS Control Region and associated Message Processing Regions, CICS, TCAM, and JES, as well as lesser tasks such as RMF and other performance monitors.

17.3. Overview of RMF Reports

In Chapter 12 we described software monitors in general. RMF is a software monitor that records information during system operation with modest overhead (typically, 2% to 3%). RMF uses a combination of *sampling* and *event recording*. For example, queue length distributions at various resources are determined by sampling, while the number of physical I/O operation (SIOs) is accumulated by event recording.

RMF generates a number of standard reports. These reports provide far more data than is needed for our purposes. Table 17.1 (which is divided into four parts) presents some relevant data items excerpted from various RMF reports for our example system. The following paragraphs briefly describe the content of the various reports.

The *CPU Activity Report*, shown in abbreviated form in Table 17.1a, provides information on the CPU and its usage. It includes the length of the observation interval, the CPU model number, the percentage of time that the CPU was idle (WAIT TIME PERCENTAGE), the average number of TSO users (TSO AVG ASIDS), and the CPU queue length distribution (not shown in the table) broken down in various ways.

Note that RMF always expresses percentages out of 100 (e.g., 32.09 rather than .3209 in Table 17.1a); we would calculate CPU busy time as:

$$B_{CPU} = \text{INTERVAL} \times (1 - (\text{WAIT_TIME_PERCENTAGE} / 100))$$

<i>CPU Activity</i>		
	INTERVAL	60 min.
	CPU MODEL	0470
	WAIT TIME PERCENTAGE	32.09%
	TSO AVG ASIDS	78.5

Table 17.1a – RMF CPU Activity Report

The *Channel Activity Report*, shown in abbreviated form in Table 17.1b, has two parts. The first reports on each physical channel, indicating its type, volume of activity (CHANNEL ACTIVITY COUNT, the total number of operations during the observation interval), and utilization, among other quantities. The second reports on each logical channel, indicating the physical channels to which it corresponds, its activity rate (measured in operations per second), and its queue length distribution (only the average queue length is shown in the table). There also is information on the percentage of requests that are delayed by congestion in the I/O subsystem, and the components responsible for such delays.

The *Direct Access Device Activity Report*, shown in abbreviated form in Table 17.1c, describes individual devices. It provides information on logical channel attachment, physical I/O count, utilization, queue length distribution (only the average queue length is shown in the table), and average service time (not shown in the table). There also is information (not shown) concerning the causes of request delays and the proportions of time that devices are used in various ways. The *Tape Device Activity Report* (not shown) provides similar information for tape devices. Note that, in contrast to the convention used throughout this book, the queue length information reported in the Device Activity Reports does *not* include the customer in service.

The *Workload Activity Report*, shown in abbreviated form in Table 17.1d, differs from the reports described so far in that it presents resource usage information broken down by performance group and performance period. In the table, group 0 (with one period) is overhead tasks, group 1 (with two periods) is batch work, and group 2 (with three periods) is TSO. Three parts of this report are shown in the table. The first part indicates the installation's service definition coefficients. As described in Section 17.2, resource consumption is reported by RMF in service units that are determined by these coefficients. The second part indicates resource consumption. The service units acquired during the observation interval by each performance group (GRP NUM) and performance period (GRP PER) are given for I/O (IOC), main storage occupancy (MSO), and CPU (CPU), as well as in total (TOT) and on a per-second basis (PER SEC). (RMF sometimes reports a fourth form of service

<i>Physical Channel Activity</i>			
CHANNEL NUMBER	CHANNEL TYPE	CHANNEL ACTIVITY COUNT	PERCENT CHANNEL BUSY
0	BYTE MPX	392	0.00
1	BLOCK MPX	130696	29.46
2	BLOCK MPX	70544	12.41
3	BLOCK MPX	77220	15.82
4	SELECTOR	64	0.00
5	BLOCK MPX	65876	12.58
6	BLOCK MPX	904	12.27
7	BLOCK MPX	130120	27.76
8	BLOCK MPX	87212	15.38
9	BLOCK MPX	70648	15.82
A	SELECTOR	23296	5.25
B	BLOCK MPX	98156	1.83
<i>Logical Channel Activity</i>			
LOG CHN	PHYS CHN	REQ PER SEC	AVG QUEUE LENGTH
1	1,7	71.1	0.56
4	2,8	42.6	1.01
6	3,9	40.2	0.22
10	5	8.4	0.19
11	5,9	5.7	0.17
12	6	0.3	0.00
17	B	5.9	0.13
(Other Logical Channels had insignificant usage.)			

Table 17.1b – RMF Channel Activity Report

acquisition for each group and period: SRB, which roughly corresponds to directly attributable CPU overhead activity.) The third part indicates, again for each group and each period, the resource consumption rate of an average transaction (AVG ABS (absorption) RATE, measured in service units per second), the number of swaps, the average number of ready requests (AVG TRANSACTS, which includes swapped-out requests so is not a good estimator of multiprogramming level), the completion count (ENDED TRANSACTS), and the response time (AVG TRANS TIME).

<i>Direct Access Device Activity</i>				
DEV ADR	LOG CHN	DEVICE ACTIVITY COUNT	% DEV BUSY	AVG Q LNTH
100	1	32796	39.1	.01
101	1	6072	9.1	.00
108	1	43912	35.5	.21
109	1	33008	42.1	.01
117	1	40136	28.6	.13
11B	1	40424	28.6	.15
1D2	1	35876	30.6	.02
298	4	6144	7.2	.00
2C1	4	18644	17.9	.13
2C8	4	20476	18.8	.24
2C9	4	16900	13.9	.07
2CB	4	18068	15.5	.08
2D3	4	10372	6.8	.02
2D8	4	10288	5.7	.02
2DB	4	5376	5.4	.01
2DC	4	23632	11.7	.03
330	6	12492	8.4	.00
332	6	8760	5.6	.06
335	6	23392	15.5	.03
339	6	34736	41.7	.01
33A	6	20288	12.7	.01
33B	6	24900	20.5	.07
33D	6	18128	14.6	.04

(Another 133 disk volumes had device busy percentages of less than 5%. The sum of their % DEV BUSY was 50.2, indicating a total of 1807 seconds of busy time.)

Table 17.1c – RMF Direct Access Device Activity Report

The *Paging Activity Report* provides a complete breakdown of paging activity from a system perspective. (Unfortunately, paging activity is *not* broken down by performance group.) A second part of this report summarizes swapping activity. It gives both logical and physical swap counts broken down by type. It also gives the total number of swaps, the swapping rates, and the average number of pages involved in each page in and page out. The *Page / Swap Dataset Activity Report* indicates the devices

Workload Activity

SERV DEF COEF:
 IOC = 5
 CPU = 10
 MSO = 3

		<-- INTERVAL SERVICE -->				
GRP NUM	GRP PER	IOC	CPU	MSO	TOT	PER SEC
000	1	62	147	42	251	70
001	1	681	966	1130	2777	771
001	2	25	109	142	276	77
001	ALL	706	1075	1272	3053	848
002	1	654	1749	416	2819	783
002	2	359	622	196	1177	327
002	3	644	1178	395	2217	616
002	ALL	1657	3549	1007	6213	1726

(IOC, CPU, MSO, and TOT service units are expressed in thousands.)

GRP NUM	GRP PER	AVG ABS RATE	NUM OF SWAPS	AVG TRANS- ACTS	ENDED TRANS- ACTS	AVG TRANS TIME (SECS)
000	1	34	4	2.00	4	.21
001	1	183	142	4.32	52	15.94
001	2	116	56	.65	12	482.18
001	ALL	174	248	4.97	64	103.36
002	1	210	24160	3.84	24292	0.46
002	2	242	836	1.34	660	8.72
002	3	353	496	1.76	356	27.54
002	ALL	253	25492	6.96	25308	1.05

Table 17.1d – RMF Workload Activity Report

used for various types of paging and swapping. Because our example model is not intended to be used for situations in which paging activity is expected to change substantially, our parameterization will not involve information from these last two reports, and they are not shown.

17.4. Customer Description

In Chapter 12 we indicated that the identification of customer classes in a model is based primarily on the workload components to be distinguished with respect to their performance.

In a queueing network model of an MVS system, a customer class sometimes represents a single performance group, sometimes a single performance period of a performance group, and sometimes an aggregation of several performance groups. For example, a TSO performance group might correspond directly to a class, or each TSO performance period might be represented as a separate class (for example, in order to be able to report response times for trivial TSO transactions). Performance groups corresponding to production batch and test batch might be aggregated into a single class.

Certain performance groups correspond to various started tasks. Some started tasks are significant workload components (e.g., CICS, IMS Control Region, IMS Message Processing Regions), and should be represented as customer classes in the model. Others can be treated as system overhead (e.g., JES, RMF, and TCAM); the resource usage of these “overhead” performance groups must be distributed carefully across the customer classes of the model.

When a customer class corresponds to two or more performance groups, the statistics in the Workload Activity Report must be aggregated. For most quantities, aggregation involves addition over the relevant performance groups. However, for those quantities that refer to a single transaction rather than to an entire performance group (AVG ABS RATE and AVG TRANS TIME are the two examples in Table 17.1d), an average weighted by throughput must be calculated. For example:

$$\text{AVG_TRANS_TIME}_{AGG} = \frac{\sum_{g \in G} \left[\text{ENDED_TRANSACTS}_g \times \text{AVG_TRANS_TIME}_g \right]}{\sum_{g \in G} \text{ENDED_TRANSACTS}_g}$$

where G is the set of performance groups that correspond to the customer class. In the rest of this chapter we will assume that such aggregation of Workload Activity Report data items has been carried out whenever necessary.

Once classes are identified and associated with performance groups, the next task is to specify the type (transaction, batch, or terminal) and workload intensity of each class. General guidelines for choosing the type of a class were given in Chapter 12. RMF treats TSO specially (in providing the average number of active terminals, for example). Consequently,

TSO can be represented as a terminal class. Without information from other sources (e.g., specialized subsystem monitors), all other classes must be treated as either batch (if the number of active tasks is known) or transaction (possibly with a memory constraint to limit the number of customers active simultaneously).

Values for workload intensity parameters can be calculated from RMF data items as suggested in Table 17.2, using data items from the CPU Activity and Workload Activity Reports. For transaction classes, the calculation is based on the assumption that throughput is equal to the arrival rate. The formula shown for batch type classes estimates the average number of ready requests residing in main memory. (As noted in Section 17.3, the RMF data item AVG TRANSACTS includes non-resident ready requests.) This formula is known to be less reliable than the other formulae presented in this chapter, often yielding a result that significantly over-estimates the actual number of “threads of control” that are concurrently active. This is an instance where “calibration” may be appropriate. One approach used frequently in practice is to represent the workload initially as a transaction class, calculating its arrival rate as ENDED_TRANSACTS / INTERVAL. After the model has been evaluated once, the workload can be converted to a batch class whose population is determined from the outputs of this initial evaluation.

<i>Transaction:</i>	
λ	$= \text{ENDED_TRANSACTS} / \text{INTERVAL}$
<i>Batch:</i>	
N	$= \frac{\text{PER_SEC_INTERVAL_SERVICE}}{\text{AVG_ABS_RATE}}$
	$= \frac{\text{total rate of service delivery}}{\text{rate of service delivery per active job}}$
<i>Terminal (TSO only):</i>	
N	$= \text{TSO_AVG_ASIDS} = \text{number of active terminals}$
Z	$= \frac{\text{TSO_AVG_ASIDS}}{\text{ENDED_TRANSACTS} / \text{INTERVAL} - \text{AVG_TRANS_TIME}}$
	$= \frac{\text{number of active terminals}}{\text{throughput}} - \text{average response time}$

Table 17.2 – Workload Intensity Parameter Value Calculation

EXAMPLE: Workload Component Identification

(from the IPS and the Workload Activity Report of Table 17.1)

We choose to treat performance group 0 as overhead. It will not be represented as a class; its resource consumption will be apportioned among the user classes.

Performance Group	Workload Component	Customer Class	Class Type
000	overhead	(none)	—
001	batch	BATCH	batch
002	interactive	TSO	terminal

EXAMPLE: Workload Intensity Calculation

(from the IPS and Tables 17.1 and 17.2)

$$N_{BATCH} = 848 / 174 = 4.9$$

$$N_{TSO} = 78.5$$

$$Z_{TSO} = \frac{78.5}{25308 / 3600} - 1.05 = 10.1 \text{ secs.}$$

TSO is assigned to a domain with a capacity of 8.

17.5. Center Description

The structure of the model is determined primarily from knowledge of the configuration.

The Device and Channel Activity Reports may reveal some system components that are so lightly utilized that they need not be included in the model. For example, of the hundreds of disk drives in a large installation, it typically is the case that less than 25% of them will have utilizations of 5% or more in any observation interval. (In our example system, 23 disks had utilizations of 5% or more, while 133 disks had utilizations of less than 5%.) Obviously, any disk with a utilization of zero can be omitted from the model. In addition, though, a single delay center can be used to represent the aggregate effect of all disks with utilizations of less than 5%. The service demand of a class at this delay center is calculated as the sum of the busy times attributed to the class at all these

disks, divided by the total number of request completions for the class. Such an aggregation reduces the amount of work involved in constructing a model “by hand”. Because queueing delays are insignificant at resources with utilizations of less than 5%, little error is introduced. When a program is used to obtain parameter values from measurement data, as described in Section 16.2.4, it is easiest to represent all devices in the model, no matter how light their use.

As recommended in Chapter 10, disk channels are represented implicitly in our example by “inflating” disk service demands to reflect path contention. Other channels are represented explicitly, while the devices to which they connect are not. The reason that other devices (e.g., tapes) need not be treated in as much detail as disks is that they do not have the same capability of concurrent activity independent of the channel. Unit record devices and their channels often are omitted from models since spooling allows their activity to be overlapped fully with other processing. Some other lightly used channels may be either omitted or represented as part of a single delay center for similar reasons.

17.6. Service Demands

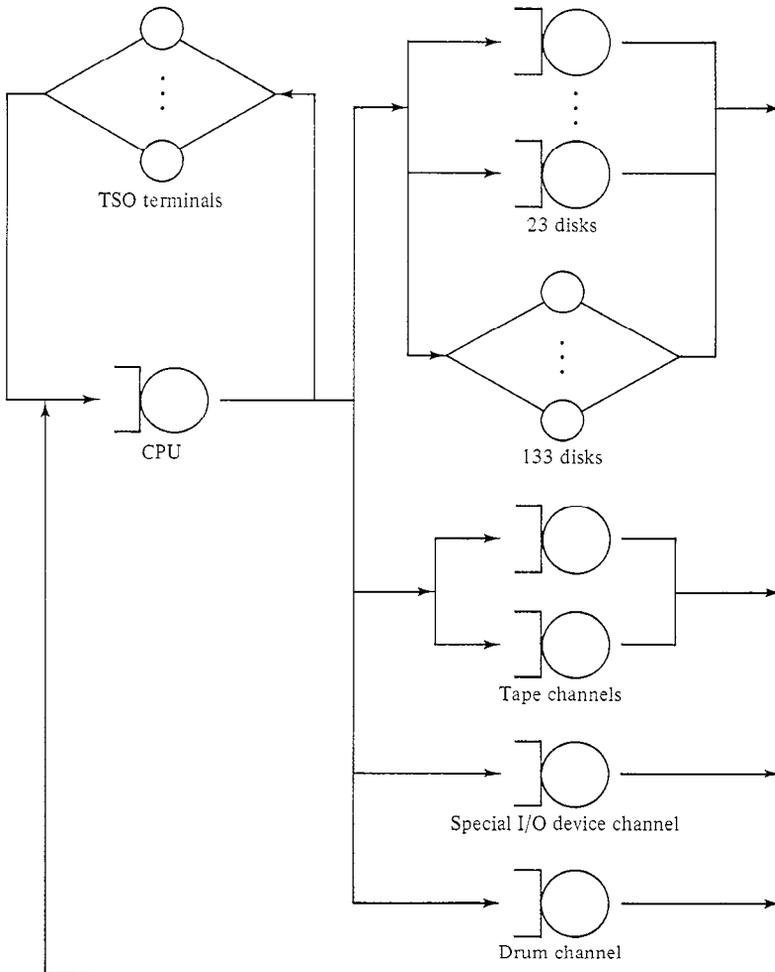
Along with the workload intensity parameters, the most critical values that must be derived from measurement data are the service demands of the customer classes at each center. The most difficult step in doing this is allocating CPU and I/O busy times to customer classes. This can be done only roughly using RMF data alone. To apportion busy times more accurately, supplementary information from other sources, e.g., the System Management Facility (SMF) or the Generalized Trace Facility (GTF), is needed. For this discussion, however, we assume that only RMF data is available.

In Chapter 12 we presented several methods for allocating unattributed CPU activity (Section 12.5.1) and I/O activity (Section 12.5.2) to customer classes. The basic quantities required by these methods are available from RMF, as shown in Tables 17.3 (CPU) and 17.4 (I/O). Note that if RMF is reporting SRB INTERVAL SERVICE (see Section 17.3), this should be added to CPU INTERVAL SERVICE in calculating attributed CPU activity. Note also that a breakdown of most physical I/Os by device and by class can be obtained by the analysis of certain types of SMF records, although we restrict ourselves to RMF here.

After attributing CPU and I/O activity to customer classes, the service demands for each class at each device are calculated by dividing the busy time attributed to a class at a center by the number of completions observed for the class (ENDED TRANSACTS).

EXAMPLE: Model Structure

# of Centers	Type	Representing
1	queueing	CPU
23	queueing	23 disks with utilization $\geq 5\%$
1	delay	133 disks with utilization $< 5\%$
4	queueing	4 channels (5, 6, A, B)



Measured CPU Busy Time:

$$B_{CPU} = \text{INTERVAL} \times (1 - (\text{WAIT_TIME_PERCENTAGE} / 100))$$

Accounted CPU Busy Time by Class:

$$A_{c,CPU} = \frac{\text{CPU_INTERVAL_SERVICE}_c}{\text{CPU_SERV_DEF_COEF} \times \text{CPU_speed_factor}}$$

where CPU_speed_factor is determined by the model.

Swapping Overhead Factor by Class:

$$SW_c = \text{NUM_OF_SWAPS}_c$$

Table 17.3 – RMF Items for CPU Activity Allocation

Disk Device Busy Time:

$$B_k = \text{INTERVAL} \times (\%_DEVICE_BUSY_k / 100)$$

Physical I/Os by Device:

$$P_k = \text{DEVICE_ACTIVITY_COUNT}_k$$

Logical I/O's by Class:

$$L_c = \frac{\text{IOC_INTERVAL_SERVICE}_c}{\text{IOC_SERV_DEF_COEF}}$$

Table 17.4 – RMF Items for I/O Activity Allocation

The two approaches to CPU activity allocation used below to treat the example system represent two extremes. While the TSO overhead factor certainly is higher than that for BATCH, it certainly is not as high as is indicated by the ratio of the reported swapping activity for the two classes.

17.7. Performance Measures

In order to validate a baseline model we need to determine from measurement data not only the input parameter values, but also the performance measure values. Table 17.5 indicates how various performance measure values can be obtained from RMF reports.

EXAMPLE: CPU Activity Allocation to Classes

First approach of Section 12.5.1; assumes that overhead is proportional to accounted usage.

$$B_{CPU} = 3600 \times (1 - .32) = 2448 \text{ secs.}$$

$$A_{BATCH,CPU} = 1075000 \times \text{factor} \quad A_{TSO,CPU} = 3549000 \times \text{factor}$$

where *factor* need not be calculated since this approach uses only relative, not absolute, accounted CPU time

$$C_{BATCH} = 64 \quad C_{TSO} = 25308$$

$$D_{BATCH,CPU} = 2448 \times \frac{1075000}{1075000 + 3549000} \times \frac{1}{64} = 8.89 \text{ secs.}$$

$$D_{TSO,CPU} = 2448 \times \frac{3549000}{1075000 + 3549000} \times \frac{1}{25308} = 0.074$$

EXAMPLE: CPU Activity Allocation to Classes

Second approach of Section 12.5.1; assumes that swapping is the primary source of overhead.

$$\text{CPU_speed_factor of Amdahl 470 V/8} = 420$$

$$\text{CPU_SERV_DEF_COEF} = 10$$

$$A_{BATCH,CPU} = \frac{1075000}{420 \times 10} = 256 \text{ secs.}$$

$$A_{TSO,CPU} = \frac{3549000}{420 \times 10} = 824 \text{ secs.}$$

$$SW_{BATCH} = 248 \quad SW_{TSO} = 25492$$

$$f_{BATCH} = 1 + \frac{\frac{248}{248 + 25492} \times [2448 - (256 + 824)]}{256} = 1.05$$

$$f_{TSO} = 1 + \frac{\frac{25492}{248 + 25492} \times [2448 - (256 + 824)]}{824} = 2.64$$

$$D_{BATCH,CPU} = \frac{1.05 \times 256}{64} = 4.2 \text{ secs.}$$

$$D_{TSO,CPU} = \frac{2.64 \times 824}{25308} = 0.086 \text{ secs.}$$

EXAMPLE: Disk Activity Allocation to Classes

In the absence of information to the contrary, assume that each class uses each disk in proportion to its overall I/O activity.

Note that the I/O activity of performance group 0 is allocated implicitly to the two classes in proportion to their accounted usage.

$$D_{BATCH_share_of_IOC_INT_SERV} = \frac{706}{706 + 1657} = .298$$

$$D_{TSO_share_of_IOC_INT_SERV} = \frac{1657}{706 + 1657} = .702$$

Calculations for disks 100 and 101 (the other 21 individually represented disks are treated similarly):

$$D_{BATCH,D100} = \frac{.298 \times .391 \times 3600}{64} = 6.576$$

$$D_{TSO,D100} = \frac{.702 \times .391 \times 3600}{25308} = 0.039$$

$$D_{BATCH,D101} = \frac{.298 \times .091 \times 3600}{64} = 1.531$$

$$D_{TSO,D101} = \frac{.702 \times .091 \times 3600}{25308} = 0.009$$

Calculations for the aggregate disk center (the total busy time of the 133 other disks is 1807 seconds):

$$D_{BATCH,DAGG} = \frac{.298 \times 1807}{64} = 8.442$$

$$D_{TSO,DAGG} = \frac{.702 \times 1807}{25308} = 0.050$$

17.8. Summary

In this chapter we have illustrated the application of the general techniques presented in Chapter 12 to a specific case. Our example treated data obtained from RMF reports concerning an MVS system. We saw that many queueing network model inputs and outputs are provided directly by RMF, while others must be calculated indirectly, with varying degrees of reliability. Similar techniques are applicable and similar difficulties are encountered in dealing with other computer systems.

EXAMPLE: Non-disk Channel Service Demands

Tape Channels (assume BATCH is responsible for all tape usage):

$$D_{BATCH,CH5} = \frac{.1258 \times 3600}{64} = 7.076 \quad D_{TSO,CH5} = 0$$

$$D_{BATCH,CHA} = \frac{.0525 \times 3600}{64} = 2.953 \quad D_{TSO,CHA} = 0$$

Electronic Drum (assume that it is used for swapping and hence should be attributed to the TSO class):

$$D_{BATCH,CHB} = 0 \quad D_{TSO,CHB} = \frac{.0183 \times 3600}{25308} = .0026$$

Special I/O Device (assume that it is used only by BATCH):

$$D_{BATCH,CH6} = \frac{.1227 \times 3600}{64} = 6.902 \quad D_{TSO,CH6} = 0$$

Throughput by Class:

$$X_c = \frac{\text{ENDED_TRANSACTS}_c}{\text{INTERVAL}}$$

Response Time by Class:

$$R_c = \text{AVG_TRANS_TIME}_c$$

CPU Utilization:

$$U_{CPU} = 1 - (\text{WAIT_TIME_PERCENTAGE} / 100)$$

Device Utilization:

$$U_k = \%_DEV_BUSY_k / 100$$

Device Queue Length:

$$Q_k = \text{AVG_Q_LNGTH}_k + (\%_DEV_BUSY_k / 100)$$

Table 17.5 – RMF Items Giving Performance Measures

EXAMPLE: Performance Measures

$X_{\text{BATCH}} = 64/3600 = .018$	$X_{\text{TSO}} = 25308/3600 = 7.03$
$R_{\text{BATCH}} = 103.4$	$R_{\text{TSO}} = 1.05$
$U_{\text{CPU}} = 68\%$	
$U_{\text{D100}} = 39.1\%$	$Q_{\text{D100}} = .40$
$U_{\text{D101}} = 9.1\%$	$Q_{\text{D101}} = .09$

17.9. References

The first published discussion of determining queueing network model parameters from measurement data was given by Rose [1978]. Since then, Levy [1980] and Irwin [1983] have presented more detailed treatments of RMF specifically. Lindsay carried out some experiments to investigate the accuracy of some of the data items obtained from RMF [Lindsay 1980].

A special issue of the the *IBM Systems Journal* concerning capacity planning contains several papers relating to the use of measurement data from MVS systems. Bronner surveys capacity planning techniques and describes the relationship among various sources of measurement data for MVS systems [Bronner 1980]. Schardt describes some techniques for using measurement data to tune MVS systems [Schardt 1980]. Schiller describes the SCAPE model, an alternative to queueing network models which is tailored to deal with many special aspects of MVS environments [Schiller 1980].

Information on MVS and its System Resources Manager (SRM) can be found in papers by Beretvas [1978] and by Lynch and Page [1974], respectively. Chiu and Chow [1978] describe a hybrid model of the internals of MVS (see Section 15.6), while Buzen describes a queueing network model based on the MVS external interface [Buzen 1978].

[Beretvas 1978]

Thomas Beretvas. Performance Tuning in OS/VS2 MVS. *IBM Systems Journal* 17,3 (1978), 290-313.

[Bronner 1980]

LeeRoy Bronner. Overview of the Capacity Planning Process for Production Data Processing. *IBM Systems Journal* 19,1 (1980), 4-27.

[Buzen 1978]

Jeffrey P. Buzen. A Queueing Network Model of MVS. *Computing Surveys* 10,3 (September 1978), 319-331.

[Chiu and Chow 1978]

Willy W. Chiu and We-Min Chow. A Performance Model of MVS. *IBM Systems Journal* 17,4 (1978), 444-462.

[Irwin 1983]

Robert T. Irwin. RMF Equations: Obtaining Job Class Results from RMF. *Journal of Capacity Management* 1,3 (1983), 230-261.

[Levy 1980]

Allan I. Levy. Introduction to Practical Operational Analysis: An MVS Perspective. *Proc. CMG XI International Conference* (1980), 208-214.

[Lindsay 1980]

David S. Lindsay. RMF I/O Time Validation. *Proc. CMG XI International Conference* (1980), 112-119.

[Lynch and Page 1974]

H.W. Lynch and J.B. Page. The OS/VS2 Release 2 System Resources Manager. *IBM Systems Journal* 13,4 (1974), 274-291.

[Rose 1978]

Clifford A. Rose. A Measurement Procedure for Queueing Network Models of Computer Systems. *Computing Surveys* 10,3 (September 1978), 263-280.

[Schardt 1980]

R.M. Schardt. An MVS Tuning Approach. *IBM Systems Journal* 19,1 (1980), 102-119.

[Schiller 1980]

D.C. Schiller. System Capacity and Performance Evaluation. *IBM Systems Journal* 19,1 (1980), 46-67.

17.10. Exercises

1. Use the multiple class mean value analysis implementation in Chapter 19 to evaluate the model for which parameters were derived in this chapter. (This will require extending the Chapter 19 implementation to handle terminal classes and delay centers. Note that you will not be able to represent the non-integer customer populations or the TSO domain capacity using this implementation.) Compare the results to the RMF performance measure values shown in Section 17.7.

2. “Calibrate” the value of N_{BATCH} as suggested in Section 17.4. (This will require extending the Chapter 19 implementation to handle transaction classes.) How are the various performance measure values affected?
3. Use Algorithm 9.2 to represent the TSO domain capacity. (This will require extending the single class mean value analysis implementation in Chapter 18 to handle load dependent service centers, as described in Chapter 20.) Note that the iteration is simplified by the fact that the batch population is fixed. How are the various performance measure values affected?
4. Based on information contained in Section 16.3, describe how you would specify this model using the queueing network modelling software package MAP.