Chapter 20

Load Dependent Service Centers

20.1. Introduction

The mean value analysis (MVA) algorithms developed in Chapters 6 and 7 allow service centers of only the queueing and delay types. As noted in Chapter 8, though, it is possible to extend these algorithms to evaluate models containing *load dependent* service centers — centers at which the service rate (the reciprocal of the service time) varies with the number of customers present. These extensions are the subject of the present appendix.

On occasion, individual components of computer systems are represented most naturally using load dependent centers. An example is a disk device where accesses are served in an order that attempts to minimize head movement. The greater the number of requests queued at such a device, the smaller the time required to satisfy each, on average, since the effectiveness of the scheduling policy increases with queue length.

The most important use of load dependent centers, though, is to implement *flow equivalent service centers* (FESCs). The construction and use of FESCs was detailed in Chapter 8, and numerous applications were noted in Chapters 9 and 11.

In discussing the modifications to MVA necessary to accommodate load dependent centers, we restrict our attention to *closed* queueing networks (batch or terminal workload types) and to the *exact* MVA algorithms (Algorithm 6.2 for the single class case and Algorithm 7.2 for the multiple class case). We begin by recalling the three principal steps of mean value analysis:

1. Compute the residence time at each center for each class, based on the service demand of the class and the average number of customers seen upon arrival to the center by a customer of that class.

2. Compute the throughput of each class as the number of customers of that class divided by the sum of its residence times at all centers (plus the think time, if the class is of terminal type).
3. Compute the queue length of each class at each center as the product of its throughput and its residence time at that center.

The exact MVA algorithms involve the iterative application of these steps at increasing populations, with the results of Step 3 at one iteration used to compute the queue lengths needed in Step 1 of the next iteration.

The load dependent versions of the algorithms involve revisions to Steps 1 and 3 — modified equations that are applied to load dependent centers:

- Consider Step 1, the estimation of the service center residence times. For load independent centers, this quantity is calculated using the (load independent) service demand and the average number of customers seen upon arrival to the center. For load dependent centers, service rates vary with queue length, so the residence time equation used in Step 1 must be augmented by terms reflecting the varying queue lengths and corresponding service rates.
- Consider Step 3, the estimation of the service center queue lengths. For load independent centers, only the average queue length is required by Step 1, so only this quantity is calculated in Step 3. For load dependent centers, the *queue length distribution* — the proportion of time that each possible customer population exists at a center — is required, so must be calculated in Step 3.

Load dependent service rates indicate the rate of customer completions at a center as a function of its current customer population. Because these rates inherently are *per visit*, while the result of the residence time equation is the total time spent at a center (i.e., the time per visit multiplied by the number of visits), service center visit counts appear as multiplicative factors in the load dependent version of the residence time equation. Thus, it appears that load dependent centers are more complicated to parameterize than load independent centers not only because of the need to give many service rates instead of a single service demand, but also because of the need to provide service center visit counts. Fortunately, this latter complication can be avoided: it is possible to rewrite the residence time equation in a way that obviates explicit visit count information. This transformation is shown in the last section of this appendix, where implementation considerations are addressed. We have chosen to include the visit count factors in the initial presentation because intuition is sacrificed in the transformation.

As in Part II of the book, our presentation is organized as a discussion of the single class case, followed by a discussion of the multiple class case. Implementation issues are discussed in a final section.

20.2. Single Class Models

We consider models with K service centers and a single customer class of batch or terminal type. Let $\mu_k(j)$ be the service rate of center k when there are j customers there. Let $p_k(j \mid n)$ be the proportion of time that center k has j customers present when the number of customers in the entire model is n. The following expressions are substituted for Steps 1 and 3 of the load independent MVA algorithm for each load dependent center k. (The load independent equations still are used for all load independent centers in the model.)

1′. Compute the residence time at load dependent center k:

$$R_k(n) = V_k \sum_{j=1}^{n} \frac{j}{\mu_k(j)} p_k(j-1 \mid n-1)$$

where V_k is the number of visits each customer makes to center k. (As noted earlier, this term is required since R_k represents the total time spent at a center, while the μ_k are service rates per visit.)

3′. Compute the queue length distribution for load dependent center k:

$$p_k(j \mid n) = \begin{cases} \dfrac{X(n)}{\mu_k(j)} p_k(j-1 \mid n-1) & j = 1, \ldots, n \\ 1 - \displaystyle\sum_{i=1}^{n} p_k(i \mid n) & j = 0 \end{cases}$$

20.3. Multiple Class Models

We consider closed, multiple class models with K service centers and C customer classes. There are two ways in which service centers in multiple class models can exhibit load dependent behavior:

- The simpler is for the service rates of all classes to vary in an identical manner as functions of the total number of customers at the center. For instance, suppose that the service rate of class A at a particular center with four customers (of any class) present is 1.5 times the service rate of class A at that center with two customers present. Then this simpler form of load dependence would require that the service rate of class B at that center with four customers present be 1.5 times its rate with two customers present.

- The more complex form of load dependence, required for the implementation of FESCs, allows the service rates of the classes to vary independently of one another, and to be functions not of the total number of customers at the center, but of the actual mix of customers there. (In this case the service center is scheduled using the fictitious *composite queueing* discipline, discussed in Chapter 8.)

We begin with the first form of load dependence. The service rate $\mu_{c,k}(j)$ indicates the rate at which class c customers would complete at center k if they were in service alone (i.e., any customers of other classes were queued but not in service) and there were a total of j customers at the center. (Again, these are rates *per visit.*) For this form of load dependence, the modifications to load independent MVA are straightforward extensions of those used in the single class case. Let the population of the model be $\vec{n} \equiv (n_1, n_2, \ldots, n_C)$, so that $n \equiv \sum_{c=1}^{C} n_c$ is the total number of customers in the model. Then for load dependent centers, Steps 1 and 3 are replaced by:

1′. Compute the residence time of class c at load dependent center k:

$$R_{c,k}(\vec{n}) = V_{c,k} \sum_{j=1}^{n} \frac{j}{\mu_{c,k}(j)} p_k(j-1 \mid \overrightarrow{n-1_c})$$

where $V_{c,k}$ is the number of visits made by each class c customer to center k.

3′. Compute the queue length distribution for load dependent center k:

$$p_k(j \mid \vec{n}) = \begin{cases} \sum_{c=1}^{C} \frac{X_c(\vec{n})}{\mu_{c,k}(j)} p_k(j-1 \mid \overrightarrow{n-1_c}) & j = 1, \ldots, n \\ 1 - \sum_{j=1}^{n} p_k(j \mid \vec{n}) & j = 0 \end{cases}$$

Now we consider the second form of load dependence, in which the service rates of each class depend on the number of customers of each class present at the center. (As explained in Section 8.4, only certain such sets of rates are valid. Further details can be found in that section.)

Let $\vec{n}$ be the customer population of the model, and let $\vec{n_k} \equiv (n_{1,k}, n_{2,k}, \ldots, n_{C,k})$ be the customer population at center k, where $n_{c,k}$ is the number of class c customers at center k. The load dependent service rates of class c at center k are denoted $\mu_{c,k}(\vec{n_k})$. As with the simpler form of load dependence, the MVA algorithm for centers of this type involves the substitution of new expressions for Steps 1 and 3 of the load independent algorithm:

1′. Compute the residence time of class c at load dependent center k:

$$R_{c,k}(\vec{n}) = V_{c,k} \sum_{\text{all } \vec{n_k}} \frac{n_{c,k}}{\mu_{c,k}(\vec{n_k})} p_k(\overrightarrow{n_k - 1_{c,k}} \mid \overrightarrow{n - 1_c})$$

3′. For each class c compute its queue length distribution at load dependent center k:

$$p_{c,k}(\vec{n_k} \mid \vec{n}) = \begin{cases} \dfrac{X_c(\vec{n})}{\mu_{c,k}(\vec{n_k})} p_k(\overrightarrow{n_k - 1_{c,k}} \mid \overrightarrow{n - 1_c}) & \vec{n_k} > \vec{0} \\ 1 - \displaystyle\sum_{\text{all } \vec{n_k} > \vec{0}} p_k(\vec{n_k} \mid \vec{n}) & \vec{n_k} = \vec{0} \end{cases}$$

20.4. Program Implementation

Fortran implementations of mean value analysis for closed models with load independent queueing centers are given in Chapters 18 and 19 for the single and multiple class cases, respectively. These programs can be modified to accommodate load dependent centers as follows:

- Alter the model definition section to allow load dependent centers to be identified and to allow load dependent rates to be provided for these centers.
- Alter the model definition section to allow service center visit counts to be provided.

As noted earlier, it is possible to rewrite the residence time equations in a way that obviates explicit visit count information, thus reducing the number of input parameters required. If this were done, the two steps outlined above would be modified. This will be discussed shortly.

- Initialize the queue length distributions at all load dependent centers for the zero population case. The distribution values should be set to one for the empty queue and to zero for all other queue populations.
- Substitute the appropriate Step 1′ for the calculation of *rtime* (statement 2001 in the Fortran programs) for each load dependent center.
- Substitute the appropriate Step 3′ for the calculation of *qlen* (statement 2003 in the Fortran programs) for each load dependent center.
- The output sections of the programs print queue lengths for each center assuming that *qlen* has been set by statement 2003. This will not be the case for load dependent centers; their average queue lengths will need to be calculated at the conclusion of the iteration, and these values assigned to *qlen*.

For many applications of FESCs, it is most convenient to avoid the specification of visit count information. Two examples of this follow:

- If the visit counts are determined by the structure of the model, they can be written into the residence time equations as constants, rather than being input by the user. For example, the techniques suggested in Section 9.3 for evaluating memory constrained queueing networks replace the central subsystem with a single FESC. It is clear that customers make one visit to this FESC per interaction, and so the visit count must be one. (See the example in Section 9.3.1.)
- Sometimes it is most convenient to define the FESC by specifying the rate at which a single customer would complete all of its service, plus a set of *service rate multipliers* that indicate the speed of the service center with a certain customer population relative to its speed with a single customer. For instance, in modelling a tightly-coupled dual processor (see Section 11.2) it is most natural to describe the processor by giving the service demand of a single customer (say, 10 seconds) and the relative rate at which instructions are executed as a function of the number of customers present (say, 1.0 with one customer and 1.8 with two or more customers). This information can be used by applying the following transformation to the residence time equations 1′ (here we show the single class case for ease of notation):

  Let the service rate multiplier for center k with j customers in its queue be denoted $\alpha_k(j)$, which is defined by:

$$\alpha_k(j) \equiv \frac{\mu_k(j)}{\mu_k(1)}$$

  Then we can rewrite the single class residence time equation 1′ as:

$$R_k(n) = \frac{V_k}{\mu_k(1)} \sum_{j=1}^{n} \frac{j}{\alpha_k(j)} p_k(j-1 \mid n-1)$$

  Since the reciprocal of the service rate with one customer in the queue is simply the service time per visit (S_k), this leads to:

$$R_k(n) = D_k \sum_{j=1}^{n} \frac{j}{\alpha_k(j)} p_k(j-1 \mid n-1)$$

  The required inputs now are the nominal service demand and the set of service rate multipliers.