# Chapter 6: Estimation with Models with One Job Class (Separable Networks)

The techniques here represent the next step up from the bounding studies of Ch. 5 in terms of model complexity and precision of outputs.

A single workload model is suitable for

- a single workload of interest, with no requirement to model the remaining workloads explicitly
- homogeneous workloads

The model is not applicable to

- multiple distinct workloads with different profiles
- investigations of consequences of changing a single class, or relative proportions of classes
- situations in which distinct results are required for distinct classes.

These methods are applicable to transaction, terminal, and batch workloads, all with fixed workload intensities.

If the intensity is not fixed in practice, model the different intensities, averaging if desired.

Several case studies are presented illustrating applications of the techniques that follow.

Case Study 1 shows the effectiveness of a very simple model in examining terminal response time as a function of the number of users. This study modeled a time sharing system that gave full use of the shared resources to one user at a time. This allowed the CPU and disk to be modeled as a single service center, eliminating the need for a separate memory queue. (1965!)

Case Study 2: a modification analysis

A notable feature of this study was the modeling of disks and channels separately. Latency and data transfer time was considered channel demand, while only seek time was charged as disk demand.

The model allows use of a disk by one user while a second user transfers data from that disk via the channel controlled by the second user. Naturally, that was not possible in the actual system. But use of the disks was fairly balanced and light, so the difference between the system and the model in this regard was a minor perturbation.

A model only has to be close enough for practical purposes.

## Case Study 3: capacity planning

Note that in this model, a saturated open system was modeled as a closed system with population equal to the maximum possible number of simultaneously active jobs.

## Solution Techniques: Open Model

The processing capacity is

$$\lambda_{sat} = \frac{1}{D_{max}}.$$

Assume the arrival rate $\lambda$ satisfies

$$\lambda < \lambda_{sat}$$

By the forced flow law, center throughput $X_k(\lambda)$ satisfies

$$X_k(\lambda) = \lambda V_k .$$

Center utilization $U_k$ satisfies

$$U_k(\lambda) = X_k(\lambda) S_k = \lambda D_k .$$

(In the case of delay centers, interpret $U_k(\lambda)$ as the average number of jobs present.)

Residence time

for delay centers $R_k(\lambda) = D_k = V_k S_k$

for queuing centers $R_k(\lambda)$ = time in service + time in queue.

The time in service is $V_k S_k = D_k$.
If $A_k(\lambda)$ = average queue length seen by a job arriving at center $k$, the average time in the queue at each visit is

$$A_k(\lambda) S_k .$$

(An assumption of the model is that the time to completion of the job in service is still $S_k$.)

So $R_k(\lambda) = V_k S_k + V_k(S_k A_k(\lambda))$.

Separability implies $A_k(\lambda) = Q_k(\lambda)$, so

$$R_k(\lambda) = D_k + D_k Q_k(\lambda)$$
$$= D_k (1 + Q_k(\lambda)).$$

By Little's Law in the context of the service center with its queue, $Q_k = \lambda R_k$.

Thus $R_k(\lambda) = D_k(1 + \lambda R_k(\lambda))$.

Solving
$$(1 - D_k\lambda) R_k(\lambda) = D_k$$
$$R_k(\lambda) = \frac{D_k}{1 - U_k(\lambda)}.$$

Putting this together

$$R_k(\lambda) = \begin{cases} D_k & k \text{ delay center} \\ \dfrac{D_k}{1 - U_k(\lambda)} & k \text{ a queuing center.} \end{cases}$$

Note $Q_k(\lambda) = \lambda R_k(\lambda) = U_k(\lambda)$ so

$$Q_k(\lambda) = \begin{cases} U_k(\lambda) & k \text{ a delay center} \\ \dfrac{U_k(\lambda)}{1 - U_k(\lambda)} & k \text{ a queuing center} \end{cases}$$

System response time, $R(\lambda)$, is

$$R(\lambda) = \sum_{k=1}^{K} R_k(\lambda).$$

The average number in the system is

$$Q(\lambda) = \lambda R(\lambda) = \sum_{k=1}^{K} R_k(\lambda).$$

Example (p. 113)

$\lambda = 0.3$ jobs/sec.

| | CPU | disk 1 | disk 2 | |
|---|---|---|---|---|
| V | 121 | 70 | 50 | |
| S | .005 | .030 | .027 | so |
| D | .605 | 2.1 | 1.35 | |

$D_{max} = 2.1$, $\lambda_{sat} = \frac{1}{2.1} \approx .476$

$(\lambda < \lambda_{sat}$ ✓ $)$

$U_{CPU}(.3) = \lambda D_{CPU} = .3(.605) \approx .182$

$R_{CPU}(.3) = \frac{D_{CPU}}{1 - U_{CPU}(.3)} = \frac{.605}{1 - .3(.605)} \approx .74$

$U_1 = .3(2.1) = .63$, $R_1 = \frac{2.1}{1 - .63} \approx 5.676$

$U_2 = .3(1.35) = .405$, $R_2 = \frac{1.35}{1 - .3(1.35)} \approx 2.269$

$R \approx .74 + 5.676 + 2.269 = 8.685$ sec./job

$Q(.3) = .3R(.3) \approx .3(8.685) \approx 2.606$ jobs

# Solution Techniques: Closed Model

These algorithms are based on three relationships.

Again taking $A_k(N)$ to be the average queue length at center $k$ encountered by an arriving job, we have

$$R_k(N) = \begin{cases} D_k & k \text{ a delay center} \\ D_k(1 + A_k(N)) & k \text{ a queuing center} \end{cases}$$

$$X(N) = \frac{N}{Z + \sum_{k=1}^{K} R_k(N)}$$

Little's Law for whole network

$$Q_k(N) = X(N) R_k(N)$$

Little's Law for individual centers

$A_k(N)$ is our obstacle. If we knew it, we could calculate the $R_k(N)$'s, hence $X(N)$ and $Q_k(N)$'s.

In the closed case, generally $A_k(N) \neq Q_k(N)$:

Consider a batch workload with $N=1$ and 2 queuing centers each with $D_i = 1$.

Then $\frac{1}{2} = U_k = Q_k$, but $A_k(1) = 0 \quad k \in \{1,2,3\}$

Separable networks have the property that $A_k(N) = Q_k(N-1)$

This gives us an iterative algorithm for computing $X(N)$, $Q_k(N)$, and $R_k(N)$.

```
for k ← 1 to K do Q_k = 0
for n ← 1 to N do
    begin
        for k ← 1 to K do
            R_k ← { D_k   for delay center
                  { D_k(1 + Q_k)   for queuing center
        X ← n / (Z + Σ_{k=1}^{K} R_k)
        for k ← 1 to K do Q_k ← X R_k
    end
```

This uses the $Q_k$ computed (or assigned) for $n-1$ to compute $R_k(n)$. Then it updates $Q_k$ to $Q_k(n)$.

The time requirement for this process is $O(NK)$. The space requirement is $O(K)$, unless you save intermediate solutions.

(In the case of an analysis of sensitivity to workload size, just save results for relevant values of $N$ in the process of calculating results for the largest $N$ of interest.)

Example: Calculate $R_k$, $X$, and $Q_k$ for a balanced batch system with $k=3$, demands $d$, and $N = 0, 1, 2, 3, 4 \ldots$

| | | $n=0$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|---|
| $D_k(1+Q_k)=R_k$: | 1 | – | $d$ | $d(1+\frac{1}{3})=\frac{4}{3}d$ | $d(1+\frac{2}{3})=\frac{5}{3}d$ | $d(1+\frac{3}{3})=\frac{6}{3}d$ |
| | 2 | – | $d$ | $\frac{4}{3}d$ | $\frac{5}{3}d$ | $\frac{6}{3}d$ |
| | 3 | – | $d$ | $\frac{4}{3}d$ | $\frac{5}{3}d$ | $\frac{6}{3}d$ |
| $\frac{n}{\sum R_k} = X$ | | – | $\frac{1}{3d}$ | $\frac{2}{4d}$ | $\frac{3}{5d}$ | $\frac{4}{6d}$ |
| $XR_k = Q_k$: | 1 | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ |
| | 2 | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ |
| | 3 | 0 | $\frac{1}{3}$ | $\frac{2}{3}$ | $\frac{3}{3}$ | $\frac{4}{3}$ |

In general

| | | $n>0$ | | $n+1$ |
|---|---|---|---|---|
| $R_k$: | 1 | $(\frac{2+n}{3})d$ | | $(1+\frac{n}{3})d = \frac{2+(n+1)}{3}d$ |
| | 2 | $(\frac{2+n}{3})d$ | | " |
| | 3 | $\frac{2+n}{3}d$ | | " |
| $X$ | | $\frac{n}{(2+n)d}$ | $= \frac{n}{(n+K-1)d}$ | $\frac{n+1}{((n+1)+K-1)d}$ |
| $Q_k$: | 1 | $\frac{n}{3}$ | | $\frac{n+1}{3}$ |
| | 2 | $\frac{n}{3}$ | | " |
| | 3 | $\frac{n}{3}$ | | " |

oh. right. duh.

In general, for K service centers and balanced demand $D_1 = D_2 = \ldots D_k = \ldots D_K$,

$Q_k(N-1) = \frac{N-1}{K}$, so

$$R_k(N) = D_k\left(1 + \frac{N-1}{K}\right) = D_k\left(\frac{N+K-1}{K}\right)$$

$$X = \frac{N}{\sum_{k=1}^{K} D_k\left(\frac{N+K-1}{K}\right)} = \frac{N}{N+K-1} \cdot \frac{1}{D_k},$$

as we used to calculate balanced system bounds (p. 90)

A numerical example of this exact MVA technique appears on p. 117. Stepping through this verifies the calculations and illustrates the technique.

The example is based on a terminal workload on a system that is modeled with a CPU, disk 1 and disk 2.

$D_{CPU} = .625$
$D_{disk1} = 2.1$
$D_{disk2} = 1.35$

$Z = 15$

Note the demands are also the response times for $n = 1$.

Exact MVA example, formulas

| | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 2 | | | n | | | |
| 3 | | k | 0 | 1 | 2 | 3 |
| 4 | Rk | CPU | - | 0.605 | $D4*(1+D8) | $D4*(1+E8) |
| 5 | | disk 1 | - | 2.1 | $D5*(1+D9) | $D5*(1+E9) |
| 6 | | disk 2 | - | 1.35 | $D6*(1+D10) | $D6*(1+E10) |
| 7 | X | | - | D3/($B$12+SUM(D4:D6)) | E3/($B$12+SUM(E4:E6)) | F3/($B$12+SUM(F4:F6)) |
| 8 | Qk | CPU | 0 | D$7*D4 | E$7*E4 | F$7*F4 |
| 9 | | disk 1 | 0 | D$7*D5 | E$7*E5 | F$7*F5 |
| 10 | | disk 2 | 0 | D$7*D6 | E$7*E6 | F$7*F6 |
| 11 | | | | | | |
| 12 | Z | | 15 | | | |

Exact MVA example, numerical results

| | | n | | | |
|---|---|---|---|---|---|
| | k | 0 | 1 | 2 | 3 |
| Rk | CPU | - | 0.605 | 0.624209 | 0.64393 |
| | disk 1 | - | 2.1 | 2.331435 | 2.60471 |
| | disk 2 | - | 1.35 | 1.445644 | 1.551185 |
| X | | - | 0.05248 | 0.103086 | 0.151516 |
| Qk | CPU | 0 | 0.03175 | 0.064347 | 0.097566 |
| | disk 1 | 0 | 0.110207 | 0.240338 | 0.394657 |
| | disk 2 | 0 | 0.070848 | 0.149026 | 0.23503 |

Z 15

# The Approximate MVA technique, closed model

benefit: the direct time dependence is $O(K)$, the number of centers. $N$ only affects the number of iterations required for convergence.

drawback: In practice, the results are close to the exact solution, but tight error bounds are not available.

The basic approach is to approximate $A_k(N)$ by a function $h$ of $Q_k(N)$

$$A_k(N) \approx h(Q_k(N))$$

A simple, useful form for $h$ is

$h(Q_k(N)) = \frac{N-1}{N} Q_k(N)$, corresponding to the heuristic that

$$\frac{Q_k(N)}{N} \approx \frac{Q_k(N-1)}{N-1} \quad \forall k$$

1. Initialize $Q_k(N) = \frac{N}{k} \quad \forall 1 \le k \le K$
2. Approximate $A_k(N)$ by $h(Q_k(N))$

3. Calculate

$$R_k(N) = \begin{cases} D_k & k \text{ a delay center} \\ D_k(1 + A_k(N)) & k \text{ a queuing center} \end{cases}$$

$$X(N) = \frac{N}{Z + \sum_{k=1}^{K} R_k(N)}$$

$$Q_k(N) = X(N) R_k(N)$$

4. If $Q_k(N)_{old}$ is close enough to $Q_k(N)_{new}$, say within 0.1 % $\forall k$, stop.

Otherwise return to step 2 with the new values of $Q_k(N)$

For example, using $N=3$ and the same data as in the previous example,

we start with $Q_{CPU}(3) = Q_{disk1}(3) = Q_{disk2}(3) = \frac{3}{3} = 1$

step 2. $A_{CPU}(3) \approx \frac{2}{3} Q_{CPU}(3) \approx 2/3$, likewise $A_{disk1}(3) \approx \frac{2}{3}$, $A_{disk2}(3) \approx \frac{2}{3}$

step 3. $R_{CPU}(3) \approx .605(1+\frac{2}{3}) \approx 1.008$

$R_{disk1}(3) \approx 2.1\ (1+\frac{2}{3}) \approx 3.5$

$R_{disk2}(3) \approx 1.35(1+2/3) \approx 2.25$

step 3 cont. $X(3) = \frac{3}{15+R(3)} \approx$

$$\frac{3}{15+1.008+3.5+2.25} \approx .1379$$

$$Q_{CPU}(3) = X(3)R_{CPU}(3) \approx .1379(1.008) \approx .1390$$

$$Q_{disk_1}(3) = X(3)R_{disk_1}(3) \approx .1379(3.5) \approx .4826$$

$$Q_{disk_2}(3) \approx .1379(2.25) \approx .3102$$

Repeat step 2

$$A_{CPU}(3) \approx \frac{2}{3}(.1390)$$

$$A_{disk_1}(3) \approx \frac{2}{3}(.4826)$$

$$A_{disk_2}(3) \approx \frac{2}{3}(.3102)$$

| Acpu | Adisk1 | Adisk2 | Rcpu | Rdisk1 | Rdisk2 | X | Qcpu | Qdisk1 | Qdisk2 | R |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.6667 | 0.6667 | 0.6667 | 1.0083 | 3.5000 | 2.2500 | 0.1379 | 0.1390 | 0.4826 | 0.3102 | 6.7583 |
| 0.0927 | 0.3217 | 0.2068 | 0.6611 | 2.7756 | 1.6292 | 0.1495 | 0.0988 | 0.4150 | 0.2436 | 5.0659 |
| 0.0659 | 0.2766 | 0.1624 | 0.6449 | 2.6810 | 1.5692 | 0.1508 | 0.0972 | 0.4043 | 0.2366 | 4.8950 |
| 0.0648 | 0.2695 | 0.1577 | 0.6442 | 2.6660 | 1.5630 | 0.1510 | 0.0972 | 0.4024 | 0.2359 | 4.8732 |
| 0.0648 | 0.2683 | 0.1573 | 0.6442 | 2.6634 | 1.5623 | 0.1510 | 0.0973 | 0.4021 | 0.2359 | 4.8700 |
| 0.0648 | 0.2681 | 0.1573 | 0.6442 | 2.6630 | 1.5623 | 0.1510 | 0.0973 | 0.4021 | 0.2359 | 4.8695 |
| 0.0648 | 0.2680 | 0.1573 | 0.6442 | 2.6629 | 1.5623 | 0.1510 | 0.0973 | 0.4021 | 0.2359 | 4.8694 |
| 0.0648 | 0.2680 | 0.1573 | 0.6442 | 2.6629 | 1.5623 | 0.1510 | 0.0973 | 0.4021 | 0.2359 | 4.8694 |
| 0.0648 | 0.2680 | 0.1573 | 0.6442 | 2.6629 | 1.5623 | 0.1510 | 0.0973 | 0.4021 | 0.2359 | 4.8694 |