Operational Analysis Based On Minimal Information

A Hypothetical Case Study

Herb Schwetman

Department of Computer Science

Purdue University

West Lafayette, Indiana 47907

CSD-TR 334

Operational Analysis Based on Minimal Information: A Hypothetical Case Study

Herb Schwetman

Department of Computer Sciences
Purdue University
West Lafayette, Indiana 47907

Introduction

An advertisement in DATAMATION (8/79, p.31, see Appendix) by the IBM Corporation described the Airlines Control Program (ACP) system used by Eastern Airlines to process passenger reservations. In the advertisement, a very brief description of this system included the following facts:

1. 6000 terminals on-line,
2. 5.6 million transactions per day,
3. 240,000 phone calls per day,
4. 155 messages per second-typical processing rate,
5. 185 messages per second-peak processing rate,
6. 2-3 second response time, at peak processing rate, and
7. 10-11 DASD (disk) accesses per message.

On reading this, one could become intrigued with this system and could attempt to analyze this system, using only the meager information given in the ad. One way of doing this is to use the methods of operational analysis developed by Denning and Buzen [DeBu78]. Using these techniques, plus one additional estimated parameter, we are able to provide several estimates of other properties of the system, including the number of disk drives requied to support the peak processing rate, the "think time" between successive messages from a single terminal, and estimate on some performance bounds.

This short note describes this analysis and gives some results for the system described. It should be noted that these results have not been validated in any way. The author has no knowledge about the real system. It would be interesting to see how these resuls compare with the actual system.

March 4, 1980

Operational Analysis

In [DeBu78], operational analysis of a computer system is described. In this article, a system is described as a network of queues. The Eastern Airlines system could probably be described by the network model shown in Figure 1.

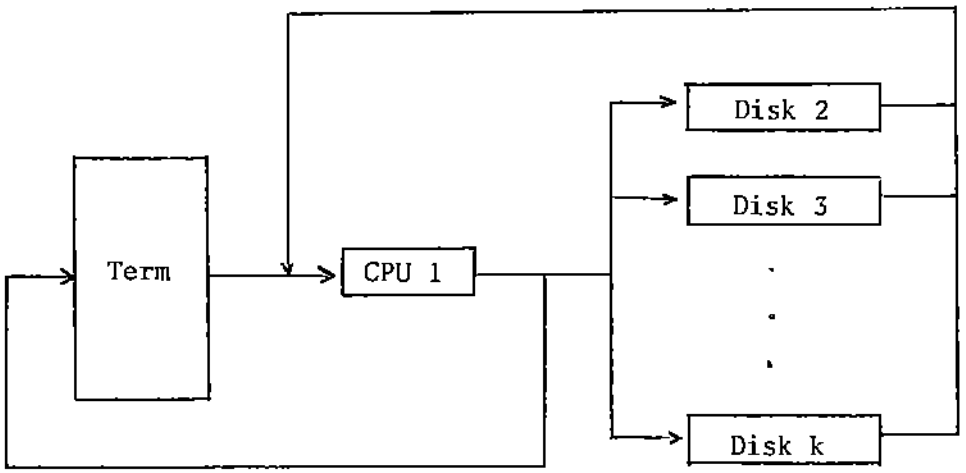


Figure 1

Schematic of System Model

Using results from operational analysis, we can state some relationships about usage of these devices while the system is operating at its maximum processing rate. More specifically, let Vi be the number of visits per message to device i and Si be the mean service time per visit to device i. Then ViSi is the total demand for device i by a message. In saturation (peak processing rate), one or more devices is always busy (call it device b) and the message processing rate for device b is 1/VbSb. It can be shown [DeBu78] that the message processing rate for the system is bounded by the maximum processing rate of the "slowest" (bottleneck) device.

Using the data from the ad we can conclude that in saturation, the message processing rate, Xo(n), is 185 messages per second, i.e., Xo(n) = 185. Thus, for some device, b, VbSb = 1/185 = .005405 sec/message. For a moment, assume that the CPU is a bottleneck device. Based on the ad, we can assume that the average message makes 10 visits to the collection of disk drives and 11 visits to the CPU (V1 = 11). Thus S1 = .005405/11 = .000491 sec. If the disks are the bottleneck, then a slightly different approach is

necessary. Let m = k-2 be the number of disk drives. If we assume that accesss to these disks are uniformly distributed (i.e., Vi = V, i = 2,...,k) and if the mean service times at each disk are equal (Si = S, i = 2,...,k), then

$$ViSi = VS = 1/185 = .005405 \text{ sec.}$$

We know that $\sum Vi = mV = 10$. Thus, if we knew S, the mean disk service time, we could then obtain an estimate for m, the number of disk drives required by the system to sustain the peak processing rate of 185 messages per second. A resonable value for a mean disk service time (seek plus latency plus transfer time) is .030 seconds. If we assume S = .030, then V = .180180 and m = 10/V = 55.5 disks drives. If m = 55 drives, then V = 10/55 = .181818, VS = .005455 and 1/VS = 183.3 messages per second. If m = 56, then V = .178571, VS = .005357 and 1/VS = 186.7 messages per second.

Thus, we can say that a model of this system which meets all of the assumptions given above and which has parameter values as shown in Table 1 would have a peak capacity of 185 messages per second.

i	0	1	2	-	57
Vi	1	11	.180180	-	.180180
Si		.000491	.030	-	.030
ViSi		.005405	.005403	-	.005403
VbSb	=	.005405			
1/VbSb	=	185			
Ro(1)	=	.305405			
Xo(1)	=	3.3			
n*	=	56.6			

Table 1

Parameter Values for System

In Table 1, Ro(1) is the mean transaction response time with one transaction in the system (Ro(1) = $\sum$ ViSi), and Xo(1) is the transaction processing rate with one transaction in the system (Xo(1) = 1/Ro(1)). The parameter n* is the number of simultaneously active transactions for which queueing at devices is certain to occur (n* = Ro(1)/VbSb). These data are presented in graphical form in Figure 2.

Table 2 shows the influence of the mean disk service time on the number of devices required to achieve the peak processing rate. As before, we assume that the pattern of accesses is uniformly distributed over all of the drives.

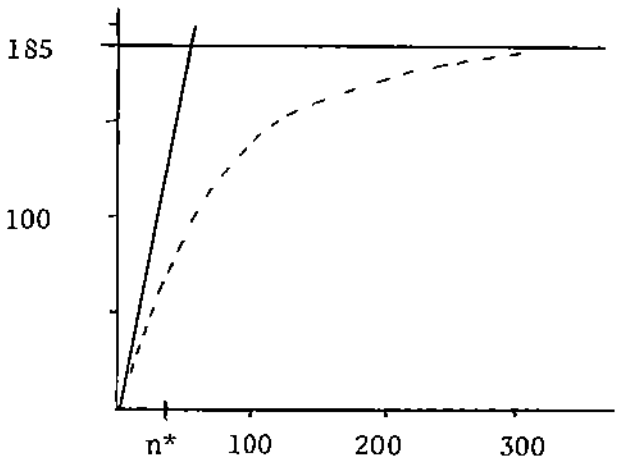


Figure 2

Plot of Throughput Bounds

Si	no. drives
.01	18.5
.02	37.0
.03	55.5
.04	74.0
.05	92.5

Table 2

Number of Drives Required
to Sustain 185 Messages per Second

Non-Uniform Disk Accesses

One possible fault with the above analysis is the assumption about uniform patterns of disk accesses. If this assumption is not valid, then it is probably because one or more of the disk drives are being accessed more frequently than others. As an example of this, assume that every transaction required one access to a directory on a single device in addition to accessing (uniformly) the other devices.

If we assume that the mean service time for the non-directory devices is again .030 seconds, then we can determine the other parameteric values which would be required for the system to achieve the peak processing rate of 185 transactions per second. Simple analysis, like that done before, shows that the directory device must have a service time of .005405 seconds and there must be about 50 non-directory disk drives.

If the directory device has a mean service interval of .010 seconds, then we need at least two directory devices, if the system is to meet the peak performance specification.

Response Time Analysis

Operational analysis can also be used to provide bounds on message response times. The model used in this analysis is shown in Figure 3.

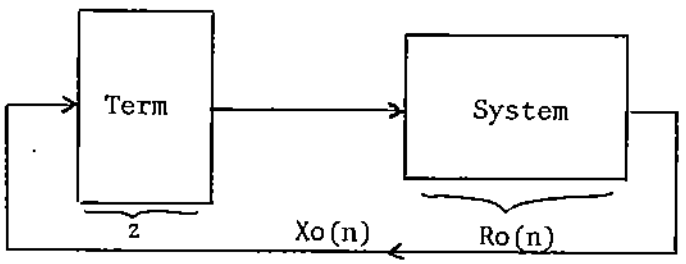


Figure 3

Response Time Model

Little's Law can be used to relate these parameters as follows:

$$n = Xo(n)(Ro(n)+Z)$$

where n is the number of active terminals, Xo(n) is the message processing rate, Ro(n) is the mean message response time and Z is the mean terminal "think" time.

The information in the ad could be interpreted in at least two ways. One way is to regard the 6000 terminals as all simultaneously active. In this case, we can use the above equation to compute the mean think time as:

$$Z = n/Xo(n) - Ro(n) = 6000/185 - 2 = 30.4 \text{ sec}$$

$$Z = 6000/185 - 3 = 29.4 \text{ sec}$$

Another interpretation is to regard 6000 as the number of terminals connectable to the system, but with some number, m, which is less than 6000 as the number of active terminals. Table 3 shows the numer of active terminals as a function of the think time for both 2 and 3 second response

times. This latter interpretation seems to be more reasonable and will be discussed in the next section.

Z	m,Ro(m)=2	m,Ro(m)=3
0	370	555
5	1295	1480
10	2220	2405
15	3145	3330
20	4070	4255
25	4995	5180
30	5920	6105
35	6845	7030

Table 3

Number of Active Terminals

A Total System Model

The facts given in the introduction, together with the comments given in the proceeding section can lead to a model of the entire system, with 6000 terminals as a part of the model. Figure 4 is a schematic diagram of this model.

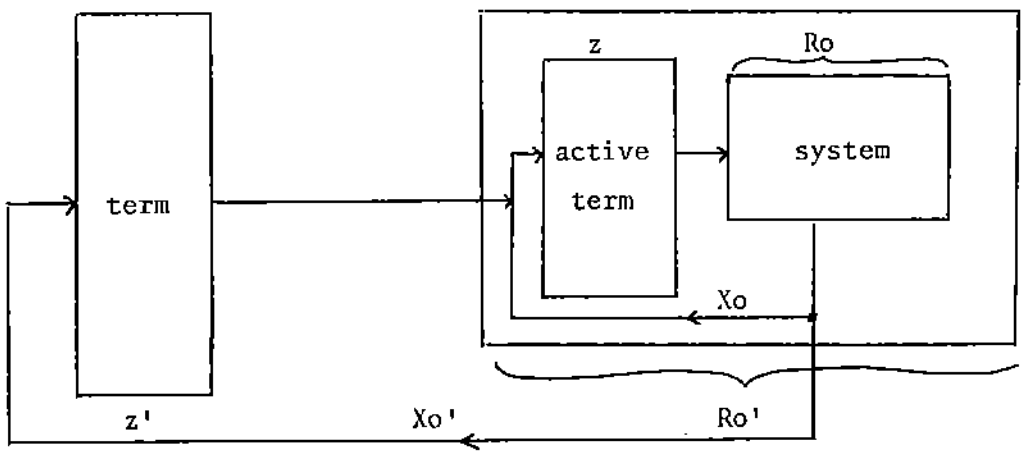


Figure 4

Schematic of System Model

In Figure 4, the Z', Xo' and Ro' are the think time, processing rate and response time variables at the level of phone calls, while Z, Xo and Ro are the same variables at the level of messages (caused by phone calls).

From the given information, we conclude that each call causes 23.3 successive transactions to be initiated. At a peak processing rate of 185 messages per second, the system can thus process 7.9 calls per second (or 476.4 calls per minute). If we assume an average think time between messages (Z) of 10 seconds, then for Ro(n) = 2 seconds, we have a per-call-response time of Ro' = 23.3(10+2) = 279.6 seconds/call (4.7 minutes/call).

Using Little's Law at the call level, we can then estimate that

a. the number of simultaneously active calls is 2220 (from Table 3), and the number of inactive terminals is 6000 - 2220 = 3780,

b. the mean time between calls (Z') is

$$Z' = 6000/7.9 - 279.6 = 479.9 \text{ sec} = 8 \text{ min.}$$

At the "typical" processing rate of 155 messages per second, Xo'(6000) is 6.7 calls per second. At this rate, if Z, the per-message think time, is 10 seconds and Ro(n) is 2 seconds, then n = 155(12) = 1860 active terminals and Z', the time between calls, becomes 615.9 seconds (10.3 minutes).

Summary

In this paper, we have attempted to describe the components of a complex system, using only the meager information found in an advertisement. Operational analysis has been the tool which has made this description a straightforward procedure.

It should be noted that this analysis used only one assumption about the stochastic (probabilistic) properties of the system, namely that the pattern of disk accesses is uniform across the disk drives, as noted above. Because of this, it is improper to use the above analysis to 'predict' performance of the system in a changed operating environment. It is possible to construct a stochastic model which has as its parameter values those observed for the actual system. If the assumptions of the model are satisfied, then it is possible to use the model to predict performance of the system.

The main value of the operational analysis approach is to use the results to gain a high level understanding of the effects of component properties and interactions on system performance, especially in a saturated mode of operation. As can be seen, this analysis is very easy to perform and requires little computational effort, even for rather

complex systems. The validity of the approach has been tested on a limited number of other systems [Schw80a, Schw80b]. As mentioned in the introduction, the author has no additional information about the Eastern Airlines system and cannot comment on the validity of the current system.

Acknowledgments

This work was first done while the author was a Fullbright-Hayes lecturer at the Deparpment of Computer Sciences, The University of Helsinki, Helsinki, Finland. In fact, some of this analysis was given as a final exam in a course taught there in the fall of 1979. The author extends his gratitude to the 16 members of that class who were "good sports" about this type of questioning.

List of References

[DeBu78] Denning, P. and J. Buzen, "Queueing Network Models of Computer System Performance", Computing Surveys (10,3), Sept. 1978, p. 225.

[Schw80a] Schwetman, H., "Operational Anaylsis - An Aid to Interpretation of Measurement Data", Computer Sciences Department, Purdue University, CSD-TR-328, January, 1980.

[Schw80b] Schwetman, H., "Modeling Performance of the B6700: A Case Study", Computer Science Department, Purdue University, CSD-TR-329, January, 1980.

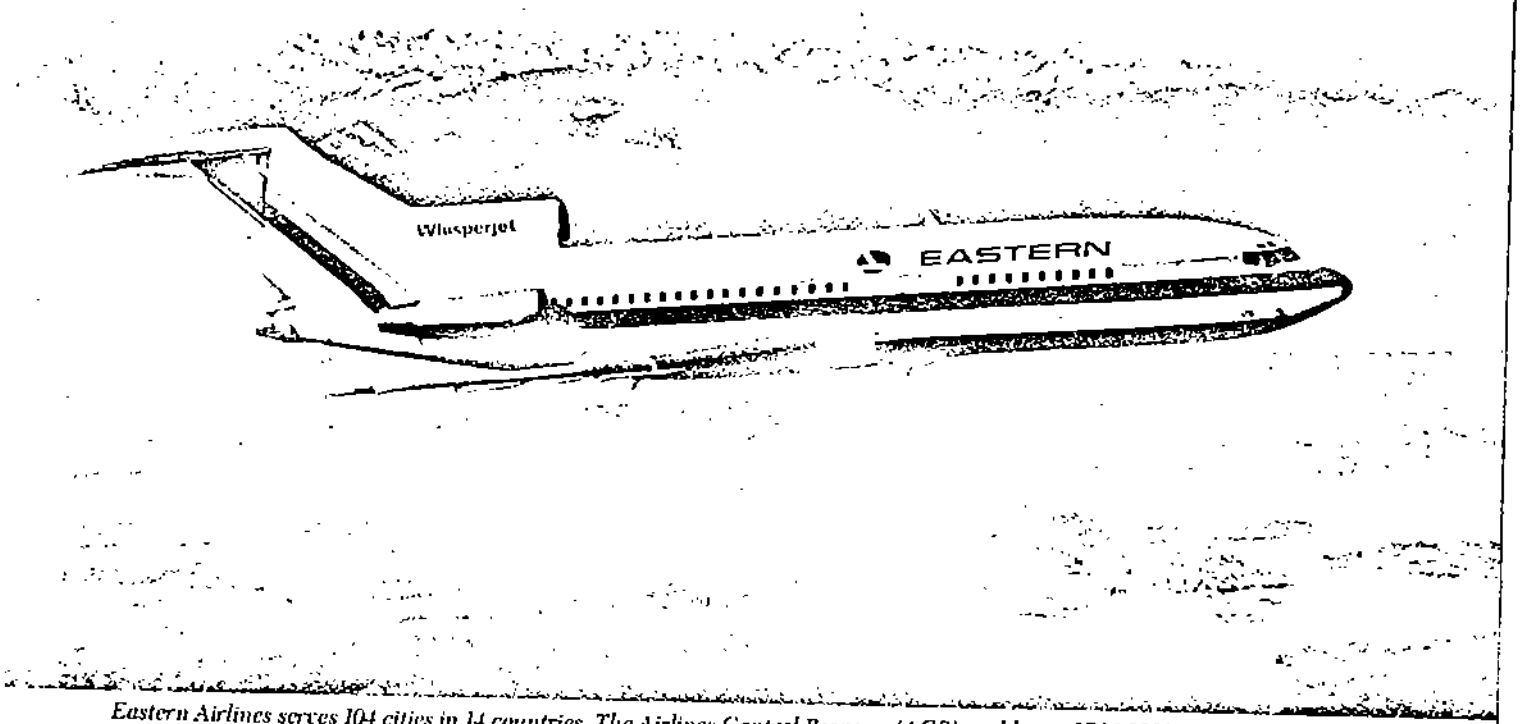
Appendix

This appendix is a copy of an advertisement which appeared in DATAMATION, August, 1979, on page 31.

Advertisement

DP Dialogue

Notes and observations from the IBM Data Processing Division that may prove of interest to DP professionals



Eastern Airlines serves 104 cities in 14 countries. The Airlines Control Program (ACP) enables an IBM 3033 Processor in Eastern's reservation system to handle 5.6 million transactions a day.

ACP Makes the Transactions Fly for Eastern Airlines

"Over 6,000 terminals are online to this computer, generating 5.6 million transactions a day," says Howard Hall of Eastern Airlines. "It often processes 155 high-speed messages per *second* – which would swamp a standard operating system. We need high-performance software and in ACP we have it."

The Airlines Control Program (ACP) is designed for IBM systems with many terminals and a high volume of transaction processing. In addition to airline reservation systems, some applications of that kind are hotel reservations, credit authorization, car rental reservations, police car dispatching, electronic funds transfer, teller memo posting, message switching, and loan payment processing.

At Eastern, agents can ask for a display of flight schedule information, seat availability, an existing passenger name file, or fare data, Hall explains. They can make reservations or change existing ones. And the system automatically computes the fare for 85 percent of the tickets issued.

System One, as Eastern calls its ACP system, also supports seat selection, boarding control, and automated ticketing at more than 100 airports, as well as flight plan calculations and a number of secondary services. Hall is director of System One, which utilizes IBM 3033 Processors in Eastern's Doral Computer Center near its corporate office in Miami, and serves 11 regional reservation centers.

"We respond to 240,000 phone calls on an average day," he notes. "So we need fast response at the terminal even at the busiest times. When ACP is running at its capacity of 185 messages a second, it still responds within two to three seconds. Of course, it is much faster than that in normal periods. And bear in mind that there are an average of 10 or 11 DASD accesses per message."

Transactions enter System One in random sequence, with many terminals competing at once for service at peak times. ACP incorporates special concepts of system control to meet these extraordinary requirements.

"Reliability is as vital as performance," Hall adds. "Much of our systemwide operation would grind to a halt without the computer. Our availability averages 99.7 percent, outside the 10 minutes a day of downtime we schedule for maintenance. And we can switch to our backup 3033 Processor in two minutes or less.

"Without ACP, System One could not possibly meet our standards of performance and reliability."