SHARE

SHARE SESSION REPORT

61	A081	MODELING FOR CAPACITY PLANNING IN A LARGE SYSTEM ENVIRONMENT	88
SHARE NO.	SESSION NO.	SESSION TITLE	ATTENDANCE

MODELING AND ANALYSIS	DAVE MUELLER	EIW
PROJECT	SESSION CHAIRMAN	INST. CODE

WAUSAU INSURANCE COMPANY, WAUSAU, WISCONSIN 54401 715-842-6636

SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER

Modeling For Capacity Planning In A Large System Environment

Steve Dodge
Pacific Telephone
120 Montgomery Street, Room 570
San Francisco, CA 94104

Modeling and Analysis Project

A081

51

03/F/TLT/2

MODELING

for

CAPACITY PLANNING

in a

LARGE SYSTEM ENVIRONMENT

Steve Dodge

Pacific Telephone (PT)

SHARE 61 Session A081

New York City

August 1983

MODELING for CAPACITY PLANNING in a LARGE SYSTEM ENVIRONMENT

OVERVIEW.

Information Systems at Pacific Telephone.

In this study, the results of a modeling effort by the MVS Capacity Planning Group in the Information Systems Organization (ISO) of Pacific Telephone will be presented. Specifically, a large CICS online system, and a large multi-CPU IMS system were modeled using the CAPTURE/MVS and BEST/1 packages from BGS Systems, Inc., of Waltham, Massachusetts.

Currently, there are four data centers in ISO - two in the northern region of California, and two in the southern. ISO runs many production IMS online systems, some of which are Centrally Developed Systems (CDS's) from Bell Labs or AT&T. In fact, in the northern region there are twelve IMS systems, six of which are CDS's. The southern region runs five production IMS systems, four of which are similar to the north's CDS's. In this study, we focus on one IMS application, which runs as two systems in each region. It is a CDS. The CICS application runs as a system in the north, and in the south, both with similar transaction volumes. This system was developed at Pacific Telephone, and is therefore not a CDS.

Capacity Planning.

The MVS Systems Capacity Planning Group in ISO was formed years ago. But, in early 1981, the make-up of the group changed, and it has remained somewhat similar up to now. During 1981, the group consisted of four people - a technical manager, an ex-systems programmer, an ex-performance analyst, and an ex-statistician. For that year, most of their concerns were with CPU capacity planning. One year later, a DASD capacity planner was added. He was an ex-computer operator.

During the period 1981 thru 1982, CPU and DASD capacity planning were done using a technique similar to that used in USAGE, from

IBM. The technique used linear projections. The greatest strides were made in the area of the materials which were presented to peer groups and upper management. A 'Systems Plan' was created, which showed workload growth in a graphic, rather than tabular, format. It was very successful. The technical manager was promoted, as was the ex-statistician. The ex-systems programmer moved on to another assignment as a manager, and the ex-performance analyst now leads the MVS Capacity Planning Group.

Techniques for capacity planning which relied on queueing theory were not used during the early years. During those two years, very little modeling of the production systems was done, because of where the priorities stood. But, in early 1983 the emphasis began to shift, and modeling began to take on a significant role. With reduced life expectancies for the new large systems hardware, it became increasingly important to improve the accuracy of the three-year applications hardware forecasts. Analytic modeling, it was thought, would fit the role. Indeed it has, and the results of two studies are presented here.

Capacity Planning vs. Performance Management.

The ISO MVS capacity planners are not performance analysts. Some experience with performance analysis tools has been greatly beneficial, but the organizational reporting hierarchy has the planners and performance persons reporting under different lines. In addition, the planners do not do performance analysis, and whatever ideas they have for improvements have never carried much clout. The planners are most concerned with making accurate six month to three year hardware forecasts. The ISO performance analysts are concerned about a much shorter time frame.

CICS MODELING.

The CICS application is seven years old, and is a locally developed system with over a hundred different task types. It is a very large application, with over 2000 terminals online in each region during production hours. The application runs on IBM 3081-K models in both regions. The northern system was modeled during April 1983, at which time it ran on a 16 Meg, 16 channel system. Seventy-three 3350 DASD were accessed during the modeling period. Two tape drives were online for logging purposes.

Inherent in any analytic modeling exercise are a number of risks and assumptions. It is most important that the analyst understand both items, especially when doing modifications or sensitivity analyses for projection to future scenarios for the application or its environment. The possible risks associated with analytic modeling used for extrapolation into the future are numerous. And, many involve unexpected changes in the application or its environment. Here, are the specific assumptions and risks involved with modeling this CICS application.

<u>Assumptions.</u>

A. 1. The CICS tasks (or transactions) can be clustered into different types. There are over 100 different task types for this application. Early in the analysis, several variables (task average CPU time, wait time, I/O time for various servers, number of terminal responses, and others) were thought to be needed to break the CICS tasks into many clusters. SAS cluster analysis was used. The results were inconclusive, because no consistent set of clusters was to be found over different measurement intervals. Instead, only two clusters were used, and they were based on wait time, not resource usage.

Cluster-1 included those tasks which had wait times less than one second. Cluster-2 included all other tasks. The CICS application has interfaces to other applications, and many of the associated interface tasks have average wait times greater than a second. All online tasks have very short wait times. By count, 93% of all tasks were Cluster-1 types.

A. 2. The dispatching priority option in BEST/1 was used, and it was assumed that Cluster-1 tasks have higher priority than Cluster-2 types. Priority modeling operates on a preemptive resume priority discipline.

A. 3. CICS ran on a 3081-D dyadic processor. It is assumed that a <u>theoretical limit</u> of one-half the system's processing time is available for CICS use, because of CICS's single address space nature. In the model, two CPU servers were

used - a server which accomodated CICS, and one which served other workloads. This is the one heroic assumption in modeling CICS.

A. 4. Memory was not a problem on the CICS 3081-D system. CICS runs in a fixed six megabyte region. There is 'no' paging. CICS is the single workload (of any importance) which runs on the system.

A. 5. In the CICS model, only basic modeling was done. No I/O subsystem modeling was attempted. In BEST/1 terminology, this assumption means that every workload transaction accesses each of the workload group's servers. The resulting model was satisfactory for CPU capacity planning, though inadequate for any performance analysis requiring I/O subsystem detail.

Now, the risks involved with modeling this CICS system.

<u>Risks.</u>

R. 1. Only a single busy hour of a single day was modeled. However, this was primarily because of CICS data set movement over several days. We just couldn't do any service time averaging for the DASD, because we didn't know where the servers were from one day to the next. So, considerable time was spent looking at SAS sysouts, to find a good modeling period. It is felt that we chose a typical measurement interval.

R. 2. Five workloads were modeled: CICS Cluster-1, CICS Cluster-2, Test Batch, TSO, and Overhead (or Other). ISO uses very little TSO, and ROSCOE is used as the card-image editor. There was a minor problem with the IPS, as ROSCOE's performance group was the same as that for the Overhead workload. ROSCOE should have a separate PG sometime soon, and then we can treat it as a separate workload.

R. 3. RMF and PAII measurement intervals were out of sync by eight minutes, or 13%. Get your measurement intervals started on the hour or half-hour!

R. 4. There was high CPU unaccounted time, which is time that is unassociated with any workload. BGS Systems tech support says that they have seen up to 20% CPU unaccounted time figures for 3081 series mainframes at some installations. Here, we had a 10% figure. How this time was distributed among the five workloads made accurate model validation of CPU utilization possible.

R. 5. There is a definite non-homogeneity to the CICS task types. But, in this analysis, it was assumed that homogeneity existed, and all tasks needed all servers. Clearly, this is not the case, as Cluster-1 tasks use different servers than Cluster-2 tasks. We couldn't separate them, though, so homogeneity is assumed.

The time involved to model the CICS system can be broken into seven phases. Although the time associated with each phase will not be the same for every modeling effort, the phases involved will be similar.

Phases.

P. 1. Choosing an application or system to model. This is easy. Getting approval may not be. For the CICS application the time involved here was insignificant.

P. 2. Establishing application or systems programming contacts and learning about their data sources. Time involved: three days.

P. 3. Gather preliminary data. One must choose an appropriate interval to model. Reading through many SAS sysouts was done during this phase. Time involved: five days.

P. 4. Run the data extractor (CAPTURE/MVS). Run the Analyzer against the extracted data. Gather any other data for the modeling interval. Pour over the sysouts to be certain of the validity of the chosen interval. Note that there was only one interval chosen in this study, so no averaging over many intervals was done. That would have lengthened considerably this phase's time. Time involved: two days.

P. 5. Run the BEST/1 model. Validate the model. This may involve combining workloads, distributing CPU unaccounted time, eliminating servers, etc. Time involved: three days.

P. 6. Run the sensitivity analyses. Vary task rates, alter the CPU processing speed, add or elimnate workloads. Analyze the results. Time involved: four days.

P. 7. Present the results. This will involve either making a viewgraph presentation or writing a report. The time involved here is variable, anywhere from a couple of days to several weeks. Here, we made a viewgraph presentation. Time involved: five days.

It is worthwhile to note that the time phases are sequential, and except for Phase 7, the number of people involved in each phase has little effect on the expected total time to complete the phases. For this effort the total time involved was about twenty-two days, or one working month. It is interesting to note that, discounting the effort involved in presenting the results, running the sensitivity analyses involved less than one-quarter of the time. Building the initial model, and validating it is where the significant amounts of time were spent.

Capacity planners involved in modeling need significant help from friendly co-workers within the company, from vendors, and from contacts in the industry. In the CICS effort we got considerable help from the CICS application's planning and programming support group. Mostly this involved learning about the makeup of the application tasks, and getting SAS source code to examine the tasks.

CICS Model Validation.

The CICS model was validated in three areas. First, the CICS application CPU utilization was validated. Second, the average internal response time for the average of the Cluster-1 and Cluster-2 tasks was validated. Last, the maximum task throughput rate was validated. Table 1 shows model validation results.

TABLE 1.

CICS Model Validation

Validation Variable	Data Source	Figure	Difference
CICS CPU Utilization	QCM	33.5%	
	Model	35.1%	4.8%
System CPU Utilization	QCM	45.5%	
	Model	45.3%	-0.4%
Average Internal Response Time	PAII	0.26 sec	
	Model	0.26 sec	0.0
Estimated Maximum Task Rate	CICS Group	90K/hour	
	Model	89K/hour	-1.1%

As can be seen from the table, system CPU utilization for the system on which the CICS application was running is in good agreement with the total obtained from the software monitor QCM, from Dusquene Systems, Inc. The total unaccounted CPU time was about 10%, and this is an important figure when validatiing CPU utilization for the application of interest. In this case, we attributed 40% of the unaccounted time to CICS, and the remainder to an Overhead workload. Generally, validation to within ten percent of the utilization figure is considered acceptable.

Table 1 also shows that the average internal response time (R.T.) for all CICS tasks is in exact agreement with Performance Analyzer II (PAII) data collected by the CICS applications group. Note that these figures are not 50th percentile R.T. figures, but rather averages of the estimated response times for all tasks. Rule of thumb says that the 95th percentile figures are going to be approximately three times as long as the average R.T. figures. A 95th percentile of one second requires an average R.T. of about 0.33 sec. Generally, Service Level Agreements list 90th or 95th percentiles for R.T. Here, we would expect 95% of all tasks to have internal response times less than one second. Also, this CICS application was sitting on the knee of the response time sensitivity analysis (RTSA) curve. When this is the case, model validation becomes difficult, so these exact response time validation results are all the more impressive.

Finally, an atypical validation parameter was used as the final check in the validation process. Long ago, an estimate of the maximum task rate for the CICS application was made by those close to the application. They estimated 90,000 tasks per hour, on 3081-D hardware as the maximum throughput rate. Here, the model shows 89,000 as the absolute maximum.

After validation, the task throughput rate was increased, such that the first bottleneck could be found. Assuming that the I/O subsystem could be tuned (dataset and pack movement), the first bottleneck which was found was the CPU. In typical modeling analysis fashion, the CPU 'SPEED' parameter was increased. The 3081-D was upgraded to a 3081-K, by using a SPEED of 1.4 - our performance factor estimate for a 3081-K is 1.4 times a 3081-G. Assuming that the one dataset which resides on a 3350 with 35% utilization was not a problem, the sensitivity analysis could proceed from there. Other bottlenecks were then found. The CICS model assumed that all DASD packs were hit by all tasks. In truth, this is simply not the case. The task mix is important. But, finding individual task service requirements from all servers is an overbearing occupation. Capacity planners want improved forecasts, foremost.

An analysis was done changing the hardware to a large uniprocessor, and those results are presented in the next section. But, for now, let is be stated that this CICS application wants to run on a large uni, as is well-known by us and the vendor community.

CICS Modeling Results.

The results of the CICS modeling experience are summarized in the four graphs which follow. Figure I.A shows CPU Utilization Sensitivity Analysis (CUSA). Figure I.B has the Response Time Sensitivity Analysis (RTSA). Figure I.C shows the I/O Bottleneck Sensitivity Analysis (IOBSA). And, Figure I.D shows 'Utilseconds' Sensitivity Analysis (USSA) curves for a situation where CPU hardware is upgraded, as a modification analysis. Utilseconds is a term we coined, and refers to the way we use the ordinate axis twice.

CUSA (Fig. I.A) can be used to estimate CPU utilization for future workloads, assuming similar configurations to that of the modeling period. In the figure, percent utilization is plotted against hourly task volume for two different CPU models. A CICS CPU threshold line is shown for the case where only one-half of the dyadic hardware is 'used' by CICS. Finally, a reference line is placed at the peak load as it existed at the time of modeling: 70,000 tasks per hour. As can be seen from the graph, 3081-D utilization at peak load is about 38 percent. Moving over to the 3081-K curve, at an equivalent percentage, the model shows a task volume of about 100,000 per hour. This is a 40% improvement in throughput, and follows directly from our estimates of the 3081-K performance factor being 1.4 times a 3081-D.

CICS CPU UTILIZATION SENSITIVITY ANALYSIS
BEST/1 MODELING APRIL 1983

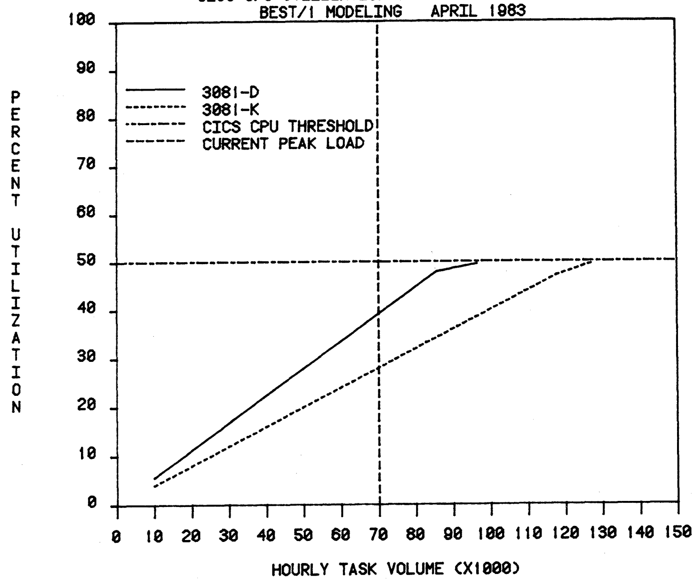

Fig. I.A
CUSA

CICS RESPONSE TIME SENSITIVITY ANALYSIS
BEST/1 MODELING APRIL 1983

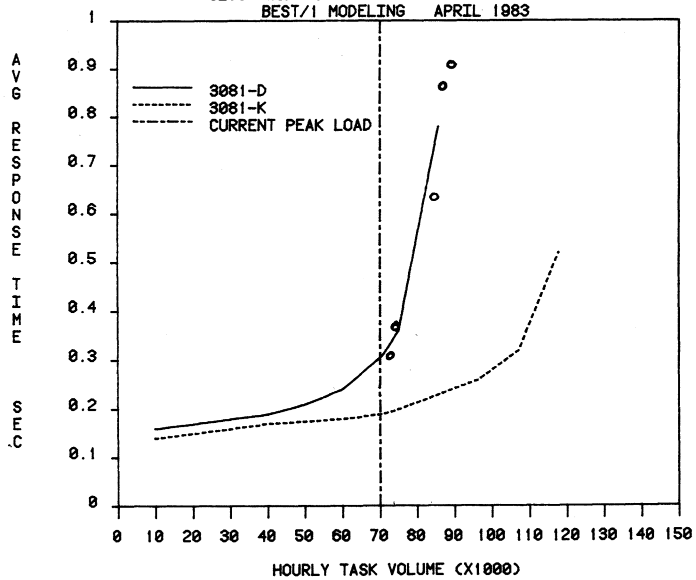

Fig. I.B
RTSA

RTSA (Fig. I.B) is used to review response time estimates for future workloads. Average response time is plotted against hourly task volume for the two different CPU models. The current peak load yields an average response time of 0.3 sec, for a 3081-D. This is slightly higher than the validation figure, 0.26, because the validation task volume was 62,000 tasks per hour. Note that a similar response time on the 3081-K projects out to about 108,000 tasks per hour. This is 54% improvement over the 3081-D. Remember that the environment as it was modeled sat directly on the knee of the response time curve, an unenviable position. There were some questions as to whether the curve was accurate. So, data for five other days has been placed on the 3081-D curve, where the task rates differed from the 70,000 figure. These data fall on the RTSA curve, and because of this it is believed that the modeled environment is one where the daily peak load falls on the knee of the curve.

IOBSA (Fig. I.C) shows percent DASD pack utilization against the hourly task volume. The two curves are for the modeling period's two high use packs, both of which were IBM 3350's. Note that when the task volume increases to the 100,000 per hour figure estimated by the CUSA and RTSA curves for the 3081-K, the first I/O bottleneck pack approaches 50% utilization. This may be a situation in which further analysis of the I/O subsystem is required. The modeling assumption which required a stable, well-tuned environment may be invalid in this high use situation. Possibly, dataset placement or redesign considerations should be looked at when a situation such as this takes place. At present, the CICS application support group is looking into a redesign of the I/O subsystem, to lower the usage of the high use pack(s).

CICS I/O BOTTLENECKS SENSITIVITY ANALYSIS
BEST/1 MODELING APRIL 1983

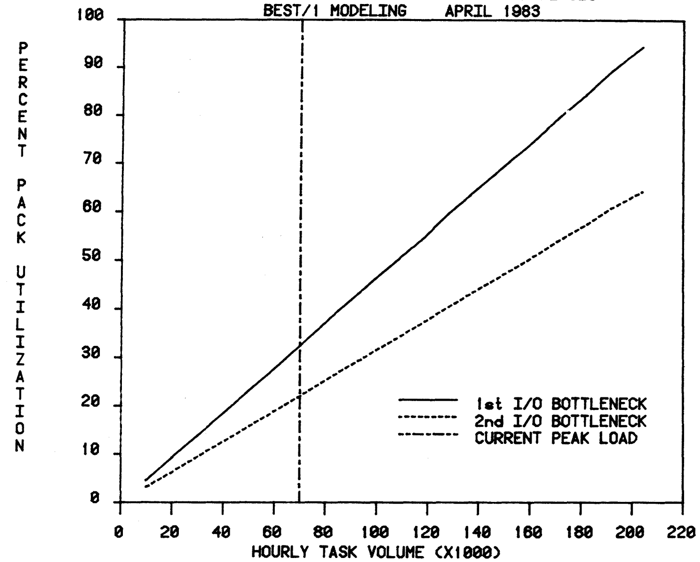

Fig. I.C
IOBSA

Finally, USSA (Fig I.D) places the CPU utilization and the response time curves on a single graph. In this scenario, a CPU modification to 2.5 times the 3081-D is made. And, because CICS is best suited for uniprocessors, a uni is assumed, so that the effective upgrade is 5.0 times one-half of the 3081-D. 'Utilseconds' is a new unit of measure, and it is plotted against the hourly task volume. Utilseconds is actually two variables rolled into one - 1) for the CPU utilization curve utilseconds represents utilization, and 2) for the response time curve utilseconds represents time in seconds. Utilseconds can be used whenever the response time is less than one second. From Fig I.D, it can be seen that the knee of the R.T. curve is beyond 160,000 tasks per hour. And, R.T. up to that point is less than 0.15 sec/task. Unfortunately, our assumption about a well-tuned environment will be invalid long before that point, and a different I/O configuration or design will have to be used.

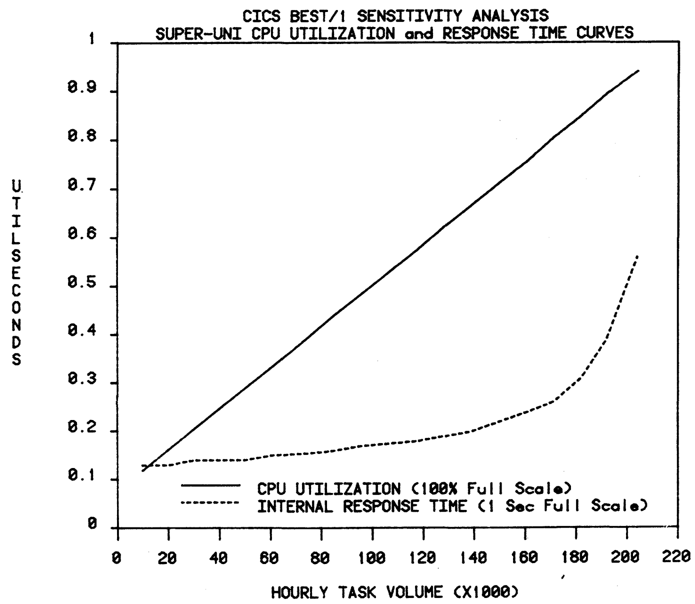

Fig. I.D
USSA

IMS MODELING.

The IMS system is about one year old, is a CDS, and currently has less than ten different transactions types. The IMS system is involved in the mechanization of part of the business, and there is considerable corporate conversion of offices to the online system. Currently, there are about 1100 VDT's in the online network. Over time, the IMS system will add many features, and will triple the number of users. For example, the system which was modeled lets users do inquiries only on the data base, whereas later releases will have update capabilities. The IMS system ran on a 24 Meg, 24 channel 3081-K, during May 1983. Dual logging was done to tape, and 103 3350 DASD volumes were accessed during the modeling period.

Here are the assumptions and risks which are associated with the modeling effort for the IMS system.

Assumptions

A. 1. The environment in which IMS ran was a well-tuned and stable environment. In fact, during the period from which modeling data was gathered, the environment was stable.

A. 2. All transactions can be processed in all MPR's (Message Processing Regions). Actually, if the number of MPR's is N, then about 90% of the transactions are processed in N-1 regions. These are 'Wait for Input' transactions. The other 10% of the transactions are normal IMS transactions, and are processed in the other MPR. The IMS system runs in conversational mode.

A. 3. Once again, we did basic modeling - no I/O subsystem modeling was done.

And, the following are the risks involved with modeling this IMS system.

Risks.

R. 1. Once again, we modeled a single busy period of a single day.

R. 2. The DC Monitor interval, used to find transaction types and volume, and the RMF/SMF extracted interval were different. The RMF/SMF interval was 30 minutes long, and the DC

Monitor interval started 10 minutes into the RMF/SMF interval, and lasted for only 10 minutes.

R. 3. There were neither BMP's nor IMS batch running during the 30 minute extraction interval. This is considered 'normal' operation during the online day. But, the worst response time occurs when the batch is running.

R. 4. The IMS control region and MPR's have different performance groups. But, when the model was built, they were grouped together into one IMS workload. They cannot be effectively modeled separately. This is a risk of no consequence.

R. 5. The CUTDEVICES facility of the CAPTURE/MVS extractor was used. There were over one hundred 3350 DASD volumes accessed by the system of interest. Fifty of them had device utilizations less than one percent. They were ignored with the CUTDEVICES command. A Delay server was added to compensate.

R. 6. Once again, high CPU unaccounted time was found on this 3081-K. And, once again, the Overhead workload got the bulk of that time.

R. 7. No log tape data appeared out of the Extractor run, and there was dual logging being done. This was because IMS bypasses the EXCP driver. The service time was estimated and the tape servers were put into the model.

R. 8. Four workloads were modeled: 1) IMS, 2) Test Batch, 3) TSO, and 4) Overhead. The TSO transaction rate was 440 trx/hour. It's CPU utilization was less than one percent.

The phases of time involvement are explained more fully in the CICS Modeling section of this report. The summary for the IMS effort follows.

Phases

P. 1. Choose an application. Time involved: insignificant.

P. 2. Establish contacts. Time involved: one day. The IMS systems programming contacts had been made previous to this effort.

P. 3. Gather preliminary data. Time involved: five days.

P. 4. Run the data extractor. Time involved: one day.

P. 5. Run the Best/1 model and validate. Time involved: three days.

P. 6. Run the sensitivity analyses. Time involved: eight days. This application has a Service Level Agreement. We ran quite a few cases of modification analysis.

P. 7. Present the results. That includes this report. The IMS section of this report took about ten working days to complete. Generally, written reports include all of the material used for viewgraph presentations, and more.

The time involved totals twenty-eight working days, so the project took about six weeks to complete. There was considerable contact with the IMS Systems Programming Group throughout the study. This was necessary because of our limited experience with IMS. Since the IMS application is a new CDS, the company's insight into the code and application may not be equal to that of the locally developed CICS application. Consequently, we relied on the IMS Systems Group.

IMS Model Validation.

The validation of the IMS model was slightly different than that for the CICS model. In CICS, one hour's extractor data was validated against one hour's QCM data, one hour's PAII data, one hour of RMF/SMF data, and an estimate of maximum task throughput rate. For the IMS model there was one-half hour of extractor data. And, it was validated against one-half hour of QCM and RMF data, and 10 minutes of DC Monitor data. That is a three to one ratio of RMF to DC Monitor intervals. Also, there was no prior estimate of the maximum IMS transaction rate.

IMS validation was done on CPU utilization, verifying against RMF and QCM data. Second a validation was done against average transaction residency time, comparing the model figure and the DC Monitor figure. BEST/1 computes average transaction residency time. The DC Monitor figure to compare against is the sum of the mean scheduling and termination time, the mean schedule to first DL/I call, and the average mean internal elapsed time per transaction. The BEST/1 figure was 0.34 sec. The DC Monitor showed

0.38 sec. The difference is -10.5%. Considering the three-to-one interval ratio, the R.T. difference is acceptable. The BEST/1 model figure for CPU utilization for the IMS application was 28%. The QCM figure was 28%. The difference is zero because of the way we distributed the CPU unaccounted time. Table 2 summarizes the validation.

TABLE 2.

IMS Model Validation

Validation Variable	Data Source	Figure	Difference
IMS CPU Utilization	QCM	28%	
	Model	28%	0.
System CPU Utilization	QCM	40%	
	Model	40%	0.
Average Transaction	DC Monitor	0.38 sec	
Residency Time	Model	0.34 sec	-10.5%

After validation, the transaction throughput rate was increased, so that the first bottleneck could be found, just as was done in the CICS modeling exercise. Once again, assuming that proper tuning on the I/O subsystem could be done, we could estimate the first system bottleneck. In this case, it was memory. So, in the sensitivity analyses which follow, memory will be varied by increasing the MPL (multi-programming level) of the IMS application.

Validating the IMS model proved to be an interesting task. There were many iterations on the Analyzer, and each iteration produced a more accurate model. The first change made to the basic model was to use the CUTDEVICES command in the Analyzer step. Of the 103 DASD devices, 50 had utilizations of less than one percent. CUTDEVICES was used to eliminate those devices. The total service time for the 50 devices was grouped into a DELAY server. This somewhat compensates for the removal of the devices. The DELAY server had the second highest utilization, 29% less than the high use pack. This produced a more accurate model than the basic model.

Next, it was discovered that the mag tape devices were not being picked up by the EXTRACTOR, so they were not found by the ANALYZER, and subsequently not found in the basic model. Dual logging was done, so two mag tape devices were put into the model

as servers. Those devices were validated on utilization figures compared against RMF figures.

Finally, CPU unaccounted time was 11%. Half of this time was distributed to the Overhead workload, and the workload CPU utilizations were validated against QCM utilization figures. This concluded the validation phase, and, the sensitivity analyses could then be performed.

IMS Modeling Results.

The IMS modeling results are summarized in the seven graphs which follow. As in the CICS modeling results, there is a CUSA graph (Fig. II.A). There are two RTSA graphs (Figs. II.B and II.C). Fig II.B is a 95% RTSA, and Fig II.C is a 90% RTSA. Unlike the CICS application, the IMS application has a SLA. There are two service level objectives for response time. The first is that in the normal environment, 95% of all transactions will have an internal response time of one second or less (Fig. II.B). The second states that in degraded mode, where there is considerable contention for resources because of an outage, 90% of all transactions will have internal response times of three seconds or less (Fig. II.C). RTSA graphs for the two SLA conditions with two different processor configurations are shown in Fig. II.D and Fig. II.E. Each figure has a curve for response time estimates for a 3081-K and another for a 3084 estimate.

The IOBSA graph for the IMS model is shown in Fig. II.F. There was only one DASD pack which showed significant utilization, so only one curve is on the IOBSA graph. The high use DASD pack is the spill area for the long message queue for IMS.

The memory sensitivity analysis (MSA) graph is shown in Fig. II.G. This is a tri-variable graph, showing probability of overcommitment of memory, the normalized memory queue, and the percent of response time spent resident in the memory queue, all plotted against hourly transaction volume. Two different multi-programming levels - four and ten - are shown.

CUSA (Fig. II.A) shows that the CPU is not a problem. IMS runs well on a dyadic processor. The transaction rate can be doubled from the current 30,000 per hour, and still be less than 60% utilization. A quad-complex, with the 3084, estimates 45% utiization at 90,000 transactions per hour. But, other servers will become bottlenecks long before the CPU.

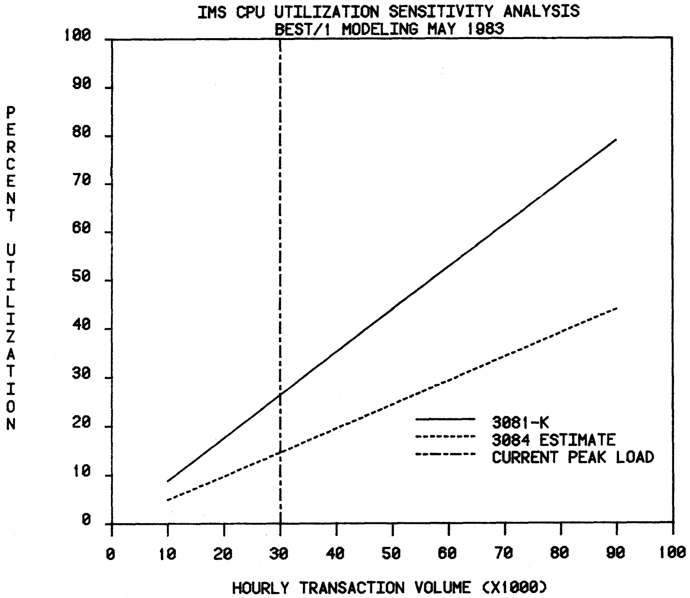

Fig. II.A
CUSA

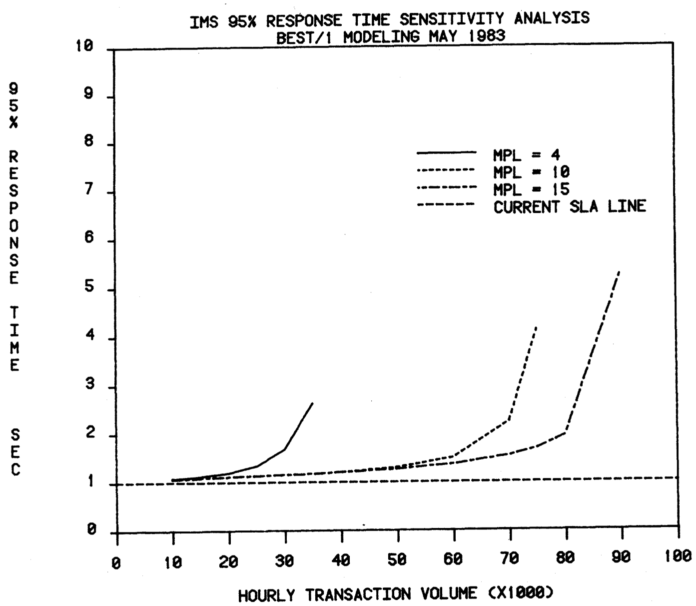

Fig. II.B
RTSA

60

The 95% RTSA (Fig. II.B) shows an interesting fact. The IMS system, as configured, sits right above the SLA response time threshold. The modeling period MPL was set at four. At this MPL (or max-MPL), the system is sitting on the knee of the response time curve. If the MPL was increased to ten, the horizontal part of the curve is slightly greater than one second, and the knee of the curve was at about 60,000 transactions per hour. But, CSA limitations are imposed, and a MPL of ten cannot be reached. A MPL of something slightly greater than four seems feasible, however. Note that a MPL of greater than 10 shows little payoff.

Since the data extracted for the modeling period was during a normal operating condition, the 90% RTSA (Fig. II.C) is not a true picture of the response time curves for the degraded mode. But, it does give a picture of how much 'response time slack' there is in the system.

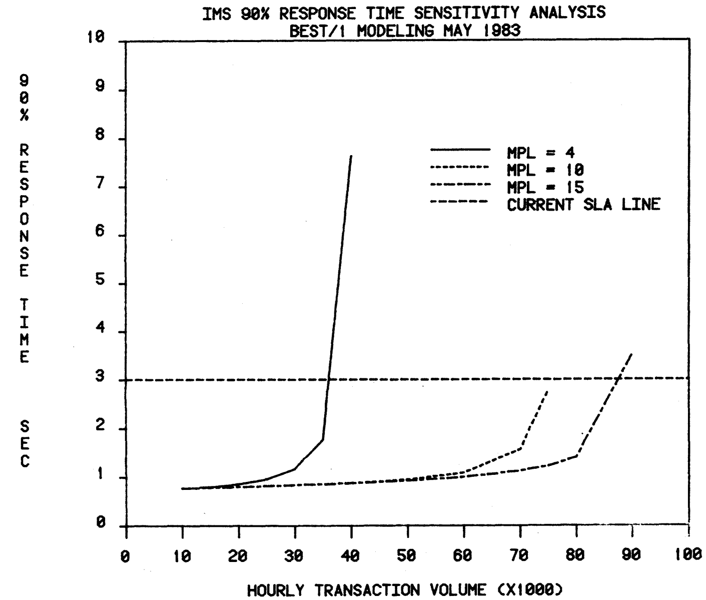

Fig. II.C
RTSA

For a 3084 CPU estimate, Fig. II.D shows that the horizontal part of the response time curve is only slightly lowered, to about one second. This shows that a majority of the response time consists of either queueing time, and/or data base access time. In fact, before it hits the knee of the curve, the response time for this IMS application is mostly data base access and wait time. Fig. II.E shows the 90% RTSA 'slack' one can expect for the degraded mode.

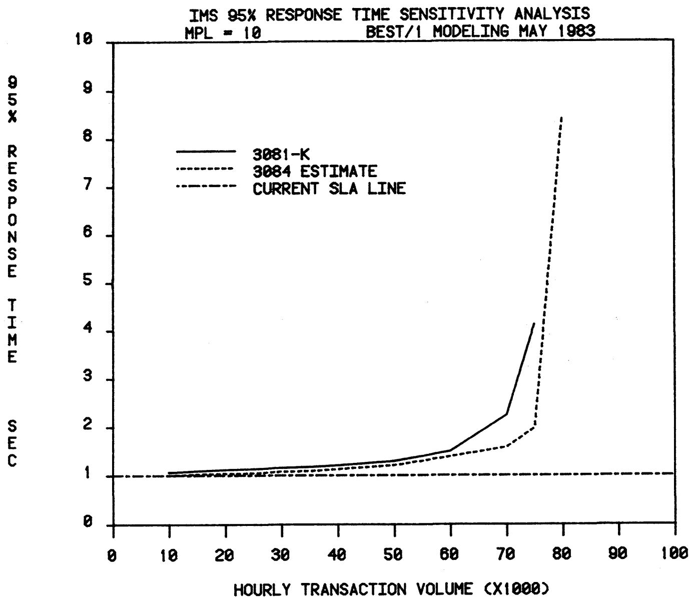

Fig. II.D
RTSA

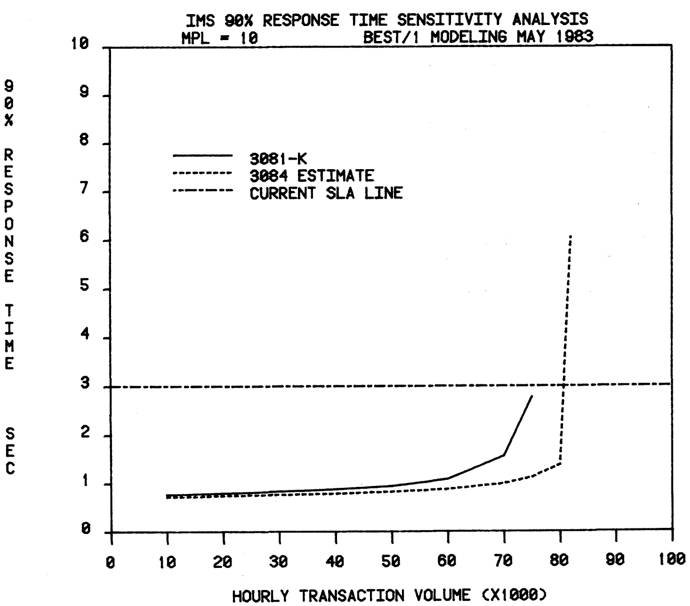

Fig. II.E
RTSA

61

IOBSA (Fig. II.F) shows that the high use pack utilization at a doubling of the current peak load to 60,000 transactions per hour is less than 45%.

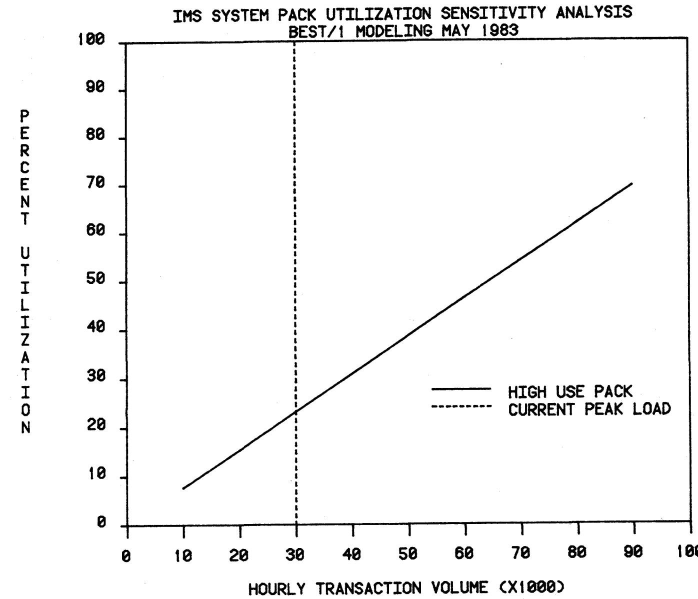

Fig. II.F
IOBSA

Finally, Fig. II.G shows the memory sensitivity analysis curve (MSA). MSA shows, for the IMS workload, three different variables graphed for two different MPL's, against hourly transaction volume. We see that all three variables represent the same characteristic - memory utilization. For both MPL's, four and ten, the following are graphed, 1) probability of overcommitment of memory, 2) the normalized memory queue, and 3) the residency time percentage spent in the input memory queue. The probability of overcommitment of memory is the approximate probability that there is at least one transaction on the input memory queue. The normalized memory queue figure is the average number of transactions on the memory queue divided by the number of MPL's. The residency time percentage spent in the memory queue plus the 'in and processing or doing IWAITs' percentage equals 100% of the residency time. Note that all three lines for both MPL=4 and MPL=10 lie in the same separate bands. Note that the current (MPL=4) peak load of 30,000 transaction per hour shows a memory figure of 0.3 (Fig. II.G). Projection to the MPL=10 band shows a throughput of 65,000 transactions per hour.

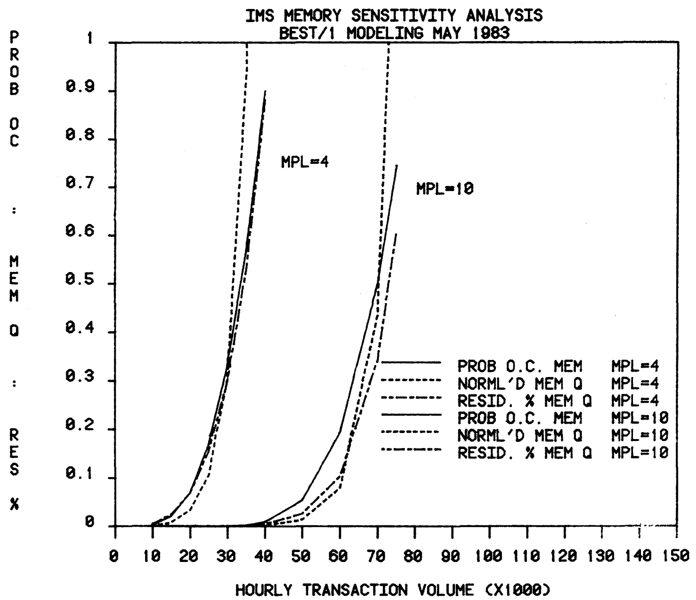

Fig. II.G
MSA

Using the CPU utilization, response time analysis, pack utilization, and memory analysis graphs, we can make projections about the resources needed to process various transaction volumes. This IMS application, as modeled, can safely process over 50,000 transactions per hour, an increase of 2/3 over the current peak load, if the maximum MPL can be increased above four.

SUMMARY.

As has been shown, a modeling exercise of a large CICS or IMS application can be valuable experience for the large system capacity planner. Deep insight into the application is a natural byproduct of the modeling experience. In addition, the analyst is given the satisfaction of knowing what are the important parameters and variables used to describe the large system. From future workoad projections, estimates of key resource consumption can be made.

Foremost in the analyst's repertoire is the understanding of the assumptions and risks associated with modeling the application and environment of interest. For all modeling efforts, the most important assumption the analyst makes is that the environment in which the application ran was well-tuned and stable during the data extraction measurement interval. Next, with large systems the analyst may need to assume that all transactions require the same resources, and the same service requirements from all servers. To do otherwise would impose heavy constraints on the analyst's time. In fact, with dynamic systems it may be impossible to break out the service requirements of each individual transaction. A major risk is that the transaction mix will change significantly. In that case, the service requirements will undoubtedly change, and a different model will need to be formed. As we have seen, there are many assumptions and risks associated with any modeling effort, and it is imperative that the analyst understand them.

In the CICS experience outlined in this study, there were a number of points made. First, for a large system, with interfaces to other systems, it may help the analyst to break out different clusters of task types, such that tasks can be scheduled on a priority basis. We found that the split between short and long wait time tasks was valuable in this modeling exercise. The short wait time tasks had a higher priority than the long ones. Next, the dyadic processor was split into two servers, one which ran CICS, and one which ran all other tasks. This allows the analyst to model an environment where total CPU utilization on the dyadic will be greater than 50%.

In the IMS large system study, we also found that a knowledge of the transaction types was valuable. The Wait for Input transactions dominate this application. We saw that the DC Monitor and the RMF monitor were not in sync, so the validation of the model was difficult. Later, we found that the log tape analysis utility can be of some use to the analyst, although we did not use it. In summary, the analyst must know which tools to use at various stages of the modeling process.

For large systems, it appears that the CUTDEVICES command of CAPTURE/MVS can be used to eliminate low utilization servers from the model. Generally, it is recommended that the analyst use any tool which simplifies the modeling requirements, without destroying the integrity of the model. CUTDEVICES is one such tool.

For capacity planners, the modeling experience provides an excellent exposure to the application variables which affect system performance. The analyst will find that he or she gains a broad knowledge of the application as it will perform under various workload and hardware scenarios. This is certainly a valuable understanding, and the analyst will benefit from it when projecting the resource requirements of the application. When making such projections, the analyst will also have a list of assumptions and limits which are applied against any sensitivity analyses. Armed with such a list, the analyst is ready to present the results of the study to management.

We have found that the best form of presentation for management is graphic in nature. Generally, graphs present considerable information in a concise manner. Tabular reports have been aggresively avoided. We believe this approach offers superior final products, with higher degrees of acceptability to management. The basic graph to management presents some dependent variable, such as utilization or response time, graphed against transaction rate. With many dependent variables to be graphed, management will find some continuity in being able to find the faithful transaction rate on the independent axis. Find what suits your immediate management, and use it to your advantage.

Acknowledgements.

The author wishes to gratefully acknowledge the following individuals for their continued support during the modeling studies: Raul Araujo, Dave Brown, Al Gonda, Fran Hildebrand, Tim Jones, Arlie Lamb, and Mike McElwee, all of Pacific Telephone.

63

SHARE

SHARE SESSION REPORT

61	A085	A Research Queuing Package (RESQ) Model of a Transaction Processing System with DASD CACHE	45
SHARE NO.	SESSION NO.	SESSION TITLE	ATTENDANCE

Modeling and Analysis	Bruce Hibbard	PJE
PROJECT	SESSION CHAIRMAN	INST. CODE

Project Software and Development, Inc., 14 Story Street, Cambridge, MA 02138
617-661-1444, Ext. 214
SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER

A Research Queueing Package (RESQ) Model of A Transaction Processing System with DASD Cache

G. A. Marazas
IBM Corporation
P. O. Box 1328
Boca Raton, FL 33431

Alvin M. Blum and Edward A. MacNair
IBM Thomas J. Watson Research Center
Yorktown Heights, New York 10598

SHARE Installation Code: IBM

SHARE Project: Modelling and Analysis

SHARE Session Number: A085

Abstract: The Research Queueing Package (RESQ) is a tool for constructing and solving models of contention systems. A contention system is a collection of interconnected resources and jobs which demand service from these resources. Examples of contention systems are computer systems, communication networks, manufacturing systems, office systems and distributed systems. We first illustrate the basic facilities available in RESQ for representing such systems and provide a simple example in order to illustrate their use.

Next we describe how RESQ has been used as an analysis tool to assist in the development of the disk cache portion of the IBM 4967 disk control unit for the IBM Series/1 computer system. The discussion here has wider application because the same design problems considered for the 4967 will also occur in one form or another in disk controllers connected to systems ranging in size from the Personal Computer to the top of the line MVS and VM systems. Also, programming design has a need similar to hardware design as to modeling and understanding sequence relationships and overlap in a complex system with many process steps. Based on such modeling experience, it is the authors' opinion that the RESQ approach involving a network of queues and the facility of passive queues is very well suited for investigation of many design issues associated with development of hardware as well as both operating systems and applications programming.

4/Y/LEJ/1