APPLICATION OF SIMULATION TECHNIQUES TO SCHEDULING AND MONITORING OF COMPUTER INSTALLATIONS

Leo J. Boelhouwer, Project Manager
Dennis M. Gilbert, Group Leader
RAAM Information Services Corporation

The Problem and the Environment

RAAM is a facilities management company. It does all of the applications design, programming and data-processing for manufacturing, distribution and retail businesses. The use of resources must be carefully budgeted to maintain a healthy financial position. It has two machines. One a System 360/Model 40, the other a System 360/Model 30. The question was posed if we could augment the Model 40 by means of extra core and multi-programming and return the Model 30 to the vendor. Savings were estimated at $15,000 a month.

Definition of the Task

Starting in February, two people spent a month defining a detailed proposal, including program and file definitions. Management responded to the proposal by establishing a goal of having an operational system capable of scheduling one day's operation on the Model 40 upon request, by June 15 of this year. Four people were allocated to the project. Work began in March on the logic design and programming. Since work was progressing on several programs simultaneously, it was necessary to test them with data simulating the output of other programs. This approach proved quite satisfactory since we successfully tested the programs as a jobstream about a week before the deadline. Then another month was needed to train the operators and control clerks to establish a definite operating procedure.

The Operational System

The result of the work is a planning model for our entire data processing facility, comprised of programming, operations and control departments. Operations include the computer room, EAM and keypunch. The control department staff is responsible for input preparation, setting up the jobs including tapes or disks so it can be handed over to the operators, and distribution of the output. The model is run each night to produce the schedule for the following day (FIG 1). It also allows management to investigate where additional equipment is needed and what effect hardware will have on overall efficiency. Conversely, if equipment goes down, the new environment can be described to the model and it will automatically refuse to schedule those jobs for which the necessary resources are not available.

Each day's expected production is specified by means of punched cards which can define an entire jobstream or a report-producing section of a jobstream or an individual program. The codes are supplied by the control clerk responsible for preparing the job and distribution of the output.

Model Capabilities

The objective was to ultimately schedule the entire data processing facility. Therefore it was necessary to schedule not only the actual running of the program but all tasks necessary from the moment the input arrives until the output leaves the premises. Any program run can generate any or all of the following jobs:

1. Control clerk activity before job is submitted
2. Keypunching of data
3. Cards read and spooled for input
4. Tape/disk mounting
5. Program run
6. Punching of output cards
7. Printout
8. Tape/disk dismounting
9. Control activity before distribution.

Every program run definition carries with it the necessary parameters to generate these sub-jobs. It also indicates which other programs have to be finished, either in the same or in another jobstream, before this program can be run. Implied pre-requisites among sub-jobs are automatically generated.

Upon request, the model can be made to select only certain types of jobs for both input and output. Thus, if only jobs which actually use the computer need to be scheduled, only they will be generated. Similarly, the output report can ignore any types of jobs that are not necessary for the user, i.e. a report for keypunch or control, showing only where they interact with the daily schedule.

Furthermore, the report program can be instructed to summarize the output to a lesser level of detail, e.g. jobstream name rather than each program, for greater ease of use (FIG 2).

It also summarized the number of hours consumed by each job type so management will know the amount of production scheduled for each type of resource and judge that it is reasonable or make adjustments.

Components

Data Base Creation

The success of the model is in no small part due to the detailed data base constructed by one member of the team. Data was gathered from run books, program documentation and interviews with programmers, control clerks and operators. The data base now contains descriptions of over 2,000 programs and over 1,000 periodic reports. It consisted of cards, three cards per record. The cards were edited and placed on tape. Future additions, deletions and changes will be edited by the same program. An update program combines the maintenance input with the existing tape and produces the updated data base.

The tape is input to a file-creation program that converts the tape file to an index-sequential disk file and inserts additional records where summary statistics may be recorded. This file is one of the inputs to the daily scheduling activity.

Retrieval of Information for Daily Scheduling

The Control Clerks submit codes for the production jobs they want to run. These cards are combined into one deck by the clerk with responsibility for running the model. She adds other 'requests' such as preventive maintenance, testing and cataloging. The retrieval program interprets the request and retrieves the pertinent records from the data base. It further generates all sub-jobs needed and supplies them with the keys of other jobs dictated by precedence relationships. It also inserts in every record the earliest time the input will be available and the deadline for completing the output as specified by the control clerk.

Establishing Latest Start Times

All the information generated by the retrieval program is passed to the clustering program, so called because it establishes clusters based on how critical the jobs are. Working backwards from a deadline it calculates the latest start time for the report producing program. This latest start time becomes the latest finish time of any program that is a direct prerequisite, which in turn reflects the time constraints to its immediate prerequisites. Finally, when a program is encountered that is the start of the jobstream, the slack is calculated as the difference between the latest start time and the earliest start time specified by the user. This slack is assigned to all members of the jobstream. Any prerequisites specified which are not members of the set of programs being scheduled are printed out with a message saying they are assumed to be complete.

Scheduling

The clusters of programs are passed to the scheduling program, where the jobs are placed in the schedule according to slack so that the most critical jobs are given highest priority. Other arbitrary parameters may be used in breaking ties, such as duration, number of I/O devices or designated priority. Every job checks when its prerequisites are scheduled to be finished and picks up from that point in time. Then the jobs already in schedule are scanned for possible conflicts. Since anything in the schedule has at least equal priority, the job is delayed if any conflicts are found. It keeps searching until an available slot is found. It is then joined to other members of the same cluster by two pointers, one pointing to the next job which starts after this one and the other to the next job which finishes after this one does. These pointers make it possible to always know the next event so that the scan can proceed in time sequence. All the jobs in progress during the earliest possible time slot in the schedule are analyzed and their requirements are added up. Whenever the scan encounters another job starting up, its resource needs are added to the total. Similarly, when a job is finished, its demands for resources are subtracted. Whenever the total requirements including those of the job being scheduled, are less than or equal to the resources available, the job can be placed in the schedule. It is assured that its requirements are met and that it will not interfere with other jobs being performed in the system.

Report System

The final program sorts the jobs by starting time so they will be back in sequence for printing. Then those program types that have been specified are printed to the desired level of summarization.

Validation of Results

The results are reviewed daily for reasonableness. Initial values of running times were highly inaccurate. As inaccuracies are encountered, new values are inserted by means of the update program. It also turned out that certain programs were never documented in the run books - the operators knew the procedures by heart. Also other programs had been deleted or changed and these changes had not been recorded. By correcting these inconsistencies, the model became gradually more reliable.

Side Benefits

The model produced a number of side benefits in addition to its use as an analytical tool. One was that the discipline of designing, programming and planning for the implementation of the model gave all involved with it an enhanced understanding of our operations even before any model output was produced and examined.

Other benefits accured from the model's data base of complete and accurate program and report descriptions. This necessitated updating of existing documentation and creation of documentation where none had previously existed. This process, in turn, caused a standardization of program descriptions including a uniform program naming convention. We also arrived at standards of minimum program documentation.

The data base allows many valuable reports to be generated, e.g., listings of programs by language, core size or any other data base parameters. These reports have facilitated conversion activities, client negotiations and job card standardization for system utilization reports.

Planned Improvements

Our data base contains two types of records, one describing the reports, another describing the programs. Every report description contains a code for calendar, so it is possible to calculate when most reports are due. Theoretically, the system can be made to determine which reports are due on any day and 'nag' until they have been produced. Similarly, it could raise an alarm when required input has not been made available. In this manner, the system could function as a monitor of all company production.

Another part of the monitoring function is to update the data base with actual run times experienced during production runs. We have programmed an interface with the DOS job accounting package which captures the run times. Before the next run of the model, these run times are reflected by means of exponential smoothing into the data base so that our predictions will be honed by experience.

Comparison to GPSS

Despite the fact that GPSS was not chosen as the vehicle for the simulation, certain functions of the model closely parallel those of certain components of GPSS. The retrieval program extracts information from the data base and generates the jobs necessary to satisfy the programs to be run. This generation of jobs produces a jobstream for the model and acts like a SPLIT block. The insertion of various parameters such as duration, deadline, earliest start time and prerequisites resemble ASSIGN block activities.

The model forms a list of jobs to be scheduled, by priority rules, and processed sequentially. This list is similar to the current events chain in GPSS.

The model also contains a collection of jobs already scheduled, grouped by slack and connected by pointers. Any job to be placed in the schedule causes a scan of all scheduled jobs in time sequence. This activity is analagous to a transaction chain scan of GPSS.

The summarizing capability of the report program which brings together jobs of the same 'family' is analogous to the GPSS ASSEMBLE block activity.

RUN DATE 10/20/71

COMPUTER SCHEDULE BY PARTITION
FOR PERIOD OF
OCT 01 1971 TO OCT 02 1971
SUMMARY AT OPTION LEVEL

COMPUTER - 360/40

SCHEDULED START HR MN	STOP HR MN	DURATN HR MN	DEDLN HR MN	DIFFERENCE HR MN	ORIG-NAME	F1 PARTITION-1	F2 PARTITION-2	BG PARTITION-3	MOD-30 PARTITION-4	NON COMPUTER
09*00	09*10	00*10	12*00	02* 50				SNY110100		
09*00	09*08	00*08	10*00	00* 52	SA300		SSA113000			
09*08	09*09	00*01	10*00	00* 52	SA304		SSA113040			
09*08	09*18	00*10	10*00	00* 42	SA307		SSA113070			
09*10	09*19	00*09	12*00	02* 41				SNY120200		
09*18	09*21	00*03	10*00	00* 39	SA700		SSA127000			
09*21	09*26	00*05	10*00	00* 34	SA310		SSA113100			
09*21	09*41	00*20	10*00	00* 19	SA230			SSA123020		
09*41	09*42	00*01	12*00	02* 19	CNTRL			SNY120210		
11*00	11*08	00*08	12*00	00* 52	SA300		SSA113000			
11*00	11*07	00*07	16*00	04* 53	SA315			SSA213150		
11*08	11*09	00*01	12*00	00* 52	SA304		SSA113040			
11*08	11*13	00*05	16*00	04* 47	SA320			SSA213200		
11*08	11*18	00*10	12*00	00* 42	SA307		SSA113070			
11*13	11*14	00*01	16*00	04* 47	SA710			SSA21S100		
11*13	11*18	00*05	16*00	04* 42	SA405			SSA214050		
11*18	11*21	00*03	12*00	00* 39	SA700		SSA127000			
11*21	11*26	00*05	12*00	00* 34	SA310		SSA113100			
11*21	11*41	00*20	12*00	00* 19	SA230			SSA123020		
12*00	12*10	00*10	17*00	04* 50	DP340		SDP513400			

RUN DATE 10/21/71

COMPUTER SCHEDULE BY PARTITION
FOR PERIOD OF
OCT 21 1971 TO OCT 22 1971
SUMMARY AT SUB-SYSTEM LEVEL

COMPUTER - 360/40

SCHEDULED START HR MN	STOP HR MN	DURATN HR MN	DEDLN HR MN	DIFFERENCE HR MN	ORIG-NAME	F1 PARTITION-1	F2 PARTITION-2	BG PARTITION-3	MOD-30 PARTITION-4	NON COMPUTER
09*00	09*10	00*10	12*00	02* 50				SNY1		
09*00	09*18	00*18	10*00	00* 42	SA300		SSA1			
09*10	09*19	00*09	12*00	02* 41				SNY1		
09*18	09*26	00*08	10*00	00* 34	SA700		SSA1			
09*21	09*41	00*20	10*00	00* 19	SA230			SSA1		
09*41	09*42	00*01	12*00	02* 19	CNTRL			SNY1		
11*00	11*08	00*08	12*00	00* 52	SA300		SSA1			
11*00	11*07	00*07	16*00	04* 53	SA315			SSA2		
11*08	11*09	00*01	12*00	00* 52	SA304		SSA1			
11*08	11*13	00*05	16*00	04* 47	SA320			SSA2		
11*08	11*18	00*10	12*00	00* 42	SA307		SSA1			
11*13	11*18	00*05	16*00	04* 42	SA710			SSA2		
11*18	11*26	00*08	12*00	00* 34	SA700		SSA1			
11*21	11*41	00*20	12*00	00* 19	SA230			SSA1		
12*00	12*19	00*19	17*00	04* 41	DP340		SDP5			
13*00	13*10	00*10	15*00	01* 50			SNY1			
13*00	13*18	00*18	16*00	02* 42	SA300			SSA1		
13*10	13*20	00*10	15*00	01* 40			SNY1			
13*20	13*28	00*08	16*00	02* 32	SA700			SSA1		
13*23	13*43	00*20	16*00	02* 17	SA230		SSA1			