Analysing the Impact of Adding a New Software System on Main Memory Usage

R. R. LEVARY* AND W. D. EDWARDS†

*Department of Management Sciences, St Louis University, St Louis, MO 63108, U.S.A.

†Citicorp Person-to-Person, Inc., St Louis, MO 63141, U.S.A.

Most computers utilise several software systems, each needing a certain memory allocation. The purpose of this study is to analyse the impact of memory allocation to different software systems on memory utilisation. Adding a new software system causes an increase in memory utilisation. Simulation is used to determine the necessary increase in memory capacity needed to accommodate the addition of a given new software system while keeping the memory utilisation at a desirable level.

A simulation of the main memory of the computer system of a large manufacturing organisation is described. The impact of adding a new software package (namely CICS) on average memory usage is analysed.

Received December 1984

1. INTRODUCTION AND SYSTEM DESCRIPTION

This article addresses the question of how an already finely tuned computer system will react to a substantial increase in its workload, and what increase in main (or primary) memory is required to return the system to its prior performance.

The operating system's scheduling algorithm is the key to the analysis of main memory usage and therefore its operation is described below. The incoming jobs are distributed on the basis of priority and placed into the backlog queue, where they remain until activated by the job scheduler. The job scheduler is the part of the operating system's control program that reads and interprets job definitions, schedules the jobs for processing, initiates and terminates the processing of jobs and job steps, and records job output data. The online transactions present themselves directly to the task scheduler which is responsible for the successive activation of the tasks making up each job.

The assignment of memory and CPU resources for the execution of computer programs is carried out by specific sections of the operating system (the dynamic allocator and CPU dispatcher). The problem of memory allocation is solved at the task level called core request queue, and core queue. Task allocation in the core request queue is a function of the initial priority for the batch jobs and the memory space required for online requests (see Ref. 2).

Paging is a memory allocation scheme that organizes memory into fixed-size blocks called page frames, and organises address space into matching-size blocks of addresses called pages. Memory is regarded as a pool of a page frames. Any page can be placed in any page frame. The page is the unit of information transmission between main (real) and auxiliary (virtual) memory. When one of the tasks in the CPU is in need of more main memory than is available, a page fault occurs and the operating system forces one of the lower priority jobs to be 'swapped out' of the CPU.

Since the application software usually consists of many core resident, reentrant modules, they should be placed into one of several categories of workload classifications. A workload classification allows an individual to describe the workload of a computer system in a minimum number of job categories. This technique reduces the effort required to produce models of a system (see Refs. 1, 3 and 5). Each category of workload classification is assigned a job scheduling priority code.

2. SIMULATION AS A TOOL FOR ANALYSING MAIN MEMORY USAGE

The first step in the design of a simulation model is the collection of relevant data. When modelling changes in an existing computer system many data are usually available since most operating systems collect some data, e.g. the OS/VS MVS Resource Measurement Facility (RMF) supplies the following information: CPU activity, workload activity, I/O device activity, paging activity, and page/swap data set activity.

To collect relevant data the system has to be studied over time. A unit of time has to be defined. Its length should be small enough for no information to be lost during this interval of time. Data on CPU activity of each software system given in CPU minutes execution time has to be converted into units of main memory usage, i.e. thousands of bytes. These data should be collected for each unit of time during a long enough length of time which ensures that the trend of memory utilisation by each software system is clearly captured.

CPU activities of software systems are random variables. Simulation of stochastic systems requires identification of the distribution of its random variables. After identifying a distribution and estimating its parameters, statistical tests, e.g. chi-square goodness-of-fit test, should be applied to find how good the theoretical distribution approximates the historical data.

3. MODEL DESCRIPTION

The simulation model is made up of three different programs, each performing a separate function.

* To whom correspondence should be addressed.

3.1. A program for simulating memory requests for workloads

Program 1 was designed to simulate memory requests of the workloads and to store the results in a workfile for use as input by the other two programs. Random numbers with the characteristics representing a given workload, i.e. type of distribution and value of its parameters, have to be generated for each workload. Methods of generating random numbers with different characteristics are described by Shannon.[6]

3.2 Simulation program for the current main memory

Program 2 takes the resource requests generated by the use of the random number generator and simulates the system's allocation of the available memory. This allocation is based on available current core storage and on the priority determined for the workload making the request.

Program 2 works in the following manner:

(1) The operating system (priority one) reads in its memory request from the workfile produced by program 1.

(2) A check is made to determine if the request can be met from the memory available.

(3) If the request can be met then the operating system is allocated the memory it requested. If the request cannot be met then the system searches for a lower class priority job (other than workloads, which can never be 'swapped out'), and takes over the memory the latter job was using. The determination of what job to reallocate memory from is based upon how active each job is in the system. The less active the job, the more likely it is to be paged. This selection has to be built into the simulation model and should be random number driven. The search for memory resources is continued until the operating system receives all the resources it requested, and then the process is repeated for the next priority workload. This repeats until all workloads have been processed.

3.3 A program for simulating the main memory after adding a new workload

Program 3 is used to illustrate the effects of memory allocation and paging when a new workload is added to the current workload environment. It has the same design and logic as program 2 except for the insertion of the new workload into a given priority. This program can be easily modified to illustrate the computer system's response to system workload for different amounts of addition to the main memory.

4. A CASE STUDY

A simulation model was developed for a large mid-western manufacturing organisation for the following reasons:

(1) To analyse the current system, i.e. to study memory allocation and paging for the current workload and memory size;

(2) To simulate the change in memory allocation (and therefore paging) caused by the addition of a Customer Information Control System (CICS) workload during the nine-hour peak between 8.00 am and 5.00 pm.

(3) To evaluate the change in main memory allocation when the (CICS) is included, in relation to the simulated increased of the main memory.

The application software was placed into one of seven categories of workload classifications. These categories have the following job-scheduling priorities listed in a descending order.

(1) Operating system (IBM MVS Release 3.8) – software that controls the execution of computer programs and can provide the following functions: (1) scheduling, (2) debugging, (3) input/output control, (4) accounting, (5) compilation, (6) storage assignment, (7) data management and related services.

(2) Environ/1 (Cincom Systems Inc., Release 8.0) – online teleprocessing monitor.

(3) CICS (IBM, Release 1.4) – Customer Information Control System; this online teleprocessing monitor is to be implemented by 1 January 1985.

(4) TSO (IBM) – Time Sharing Option, an option of the operating systems MVT and OS/VS2 that provides conversational time sharing from remote terminals.

(5) Editor (in-house) – online program development.

(6) Batch Production – batch jobs that are necessary to meet the daily requirements of the company.

(7) Batch Test – batch production jobs in the process of development and testing.

The pertinent data gathered during three months were broken down into daily nine-hour peak periods, and converted from total CPU minutes execution time to main memory usage. By observing the relative frequency of the data (not shown here) it was determined to be normally distributed. The mean and standard deviations were then calculated for each workload during this period. These values are given below in units of thousands of bytes:

Operating System, 1937, 206;
Environ/1, 2900, 503;
TSO, 334, 69;
Editor, 514, 192;
Batch Production, 2162, 754;
Batch Test, 1540, 313.

These values proved to be ineffective because they represented the average memory usage during a nine-hour peak period, instead of the memory usage at any given time during the nine-hour peak period. Observations of memory usage were then made at eighteen-minute intervals during the nine-hour peak period for one month. The relative frequencies of memory usage for all time intervals indicated that all are normally distributed. These time intervals were used for all memory simulations described below.

The normally distributed random numbers were generated for each workload by the use of the following well-known approximation method [4]:

$$Y(i,j) = \left[\sum_{k=1}^{12} R_k - 6 \right] X(i,j) + S(i,j),$$

where $Y(i, j)$ = normally distributed random number representing workload type j during time interval i, i = index representing an 18 minute time interval ($i = 1, 2, 3, \ldots, 30$), j = index representing the type of workload ($j = 1, 2, 3, \ldots, 7$), R_k = a uniformly distributed random number, $0 \leqslant R(k) < 1$, $X(i, j)$ = the mean of workload j during time interval i, $S(i, j)$ = the standard deviation of workload j during time interval i.

Since the random numbers generated by this approximation method did not adequately represent the normal

distribution of the workloads, the following modification was made: 100 generated values for $Y(i, j)$ were averaged to produce a single random number which is used to represent workload type j during time interval i. A chi-square goodness-of-fit test at the 0.05 significance level concluded that these random numbers adequately approximated the normal distributions in question.

The simulation program for the current main memory was based on 12000000 bytes of core storage. This program assumes that Environ/1 can never be 'swapped out'. The program for simulating the main memory after adding a new workload is based on a CICS software as the new workload.

5. SIMULATION RESULTS

An important objective of the managers of data processing operations in the organisation in question is to keep the average daily peak load CPU usage under 80%. This policy allows for some reserves to be used during heavy computer usage periods. This objective is currently satisfactorily met.

The simulated workload created by program 1 is given in Table 1. This table specifies that at time period N, the workload components will request x amount of main memory. Whether or not their requests can be met by the resources available is determined by simulation programs 2 and 3. Simulation results of current memory usage based on 12000000 bytes of available CPU storage, without CICS added, and with CICS added are given in Tables 2 and 3 respectively.

Table 1. Simulated workload

Time period	Operating system	Environ/1	CICS	TSO	Editor	Batch production	Batch test
1	01961	03947	02175	00917	00372	01119	01351
2	01877	02691	02081	01055	00382	00951	01672
3	01790	03317	02588	00608	00210	01060	01835
4	01864	02946	02374	00969	00416	02241	01388
5	01633	04331	01910	00715	00377	02147	01407
6	01652	02229	01578	00971	00336	01885	01632
7	01862	02359	01811	00911	00396	02721	01641
8	01836	02678	01614	01213	00494	01715	01357
9	01639	02297	02265	00952	00412	02216	00984
10	01646	03083	02592	00887	00390	04200	01009
11	01480	03464	02689	01025	00443	01155	01428
12	01795	03285	02404	00912	00380	01685	01019
13	01914	02894	01663	01152	00448	00766	01822
14	01648	02763	03196	00835	00446	01842	01173
15	01540	02864	02390	00821	00623	00490	01895
16	01790	02962	01817	00968	00463	01656	02121
17	01835	02685	01726	00832	00262	01965	01532
18	01853	02401	02372	01204	00352	01867	01783
19	01679	02982	02372	01021	00175	02020	01271
20	01332	03800	01899	00992	00349	03182	01424
21	01455	03089	02334	01064	00433	01947	01230
22	02048	03425	02462	01079	00316	01213	01670
23	01957	02646	02289	00803	00395	01504	02148
24	01445	02691	02594	01013	00427	00628	01305
25	01659	04266	02027	01115	00559	03789	01226
26	01477	02724	01769	00931	00424	02525	01787
27	01486	03004	01995	01156	00343	01300	01904
28	01507	02685	02103	00809	00420	02517	01445
29	01955	02766	02451	00934	00426	00744	01184
30	01700	02082	02628	01053	00400	02358	01148

Table 2. Simulation of current CPU usage without CICS added based on 12000000 bytes of available CPU storage

Time period	Operating system	Environ/1	TSO	Editor	Batch Production	Batch test	Idle time	% CPU usage
1	1961	3947	917	372	1119	1351	2333	0.80
2	1877	2691	1055	382	951	1672	3372	0.71
3	1790	3317	608	210	1060	1835	3180	0.73
4	1864	2946	969	416	2241	1388	2176	0.81
5	1633	4331	715	377	2147	1407	1390	0.88
6	1652	2229	971	336	1885	1632	3295	0.72
7	1862	2359	911	396	2721	1641	2110	0.82
8	1836	2678	1213	494	1715	1357	2707	0.77
9	1639	2297	952	412	2216	984	3500	0.70
10	1646	3083	887	390	4200	1009	785	0.93
11	1480	3464	1025	443	1155	1428	3005	0.74
12	1795	3285	912	380	1685	1019	2924	0.75
13	1914	2894	1152	448	766	1822	3004	0.74
14	1648	2763	835	446	1842	1173	3293	0.72
15	1540	2864	821	623	490	1895	3767	0.68
16	1790	2962	968	463	1656	2121	2040	0.83
17	1835	2685	832	262	1965	1532	2889	0.75
18	1853	2401	1204	352	1867	1783	2540	0.78
19	1679	2982	1021	175	2020	1271	2852	0.76
20	1332	3800	992	349	3182	1424	921	0.92
21	1455	3089	1064	433	1947	1230	2782	0.76
22	2048	3425	1079	316	1213	1670	2249	0.81
23	1957	2646	803	395	1504	2148	2547	0.78
24	1445	2691	1013	427	628	1305	4491	0.62
25	1659	4266	1115	559	3789	612	0	1.00
26	1477	2724	931	424	2525	1787	2132	0.82
27	1486	3004	1156	343	1300	1904	2807	0.76
28	1507	2685	809	420	2517	1445	2617	0.78
29	1955	2766	934	426	744	1184	3991	0.66
30	1700	2082	1053	400	2358	1148	3259	0.72

Average CPU usage: 77%.

Notes: (1) The data is given in thousands of bytes. (2) A time period represents 18 minutes during the nine-hour peak shift: 8.00 am to 5.00 pm.

The average CPU usage (77%) found from the actual data (not shown here) is identical to that found from the simulated CPU usage given in Table 2. This assures a valid simulation model. By comparing Table 1 and Table 2 it is concluded that in the simulation of the current system only one memory allocation request was not met (requested by batch test during time period 25). In other words, only one occurrence of demand paging caused by insufficient memory resources in the simulated environment was identified, suggesting a highly efficient computer system.

Table 3 illustrates the drastic increase in paging caused by the addition of the CICS workload to the current environment (i.e. 12000000 bytes of memory). By comparing Tables 2 and 3 the profound decrease in performance of the lower-priority jobs is clearly noticeable. The simulated average CPU usage of 93% is unacceptable.

The effect of the additional CICS workload under increasing the capacity of the main memory was simulated by modifying the available memory in program 3. Although increasing the main memory by

Table 3. Simulation of current CPU usage with CICS added based on 12000000 bytes of available CPU storage

Time period	Operating system	Environ/1	CICS	TSO	Editor	Batch production	Batch test	Idle time	% CPU usage
1	1961	3947	2175	917	372	1119	1351	158	0.98
2	1877	2691	2081	1055	382	951	1672	1291	0.89
3	1790	3317	2588	608	210	1060	1835	592	0.95
4	1864	2946	2374	969	416	2241	1190	0	1.00
5	1633	4331	1910	715	377	2147	36	851	0.92
6	1652	2229	1578	971	336	1885	1632	1717	0.85
7	1862	2359	1811	911	396	2721	1641	299	0.97
8	1836	2678	1614	1213	494	1715	1357	1093	0.90
9	1639	2297	2265	952	412	2216	984	1235	0.89
10	1646	3083	2592	887	390	2216	1009	177	0.98
11	1480	3464	2689	1025	443	1155	1428	316	0.97
12	1795	3285	2404	912	380	1685	1019	520	0.95
13	1914	2894	1663	1152	448	766	1822	1341	0.88
14	1648	2763	3196	835	446	1842	1173	97	0.99
15	1540	2864	2390	821	623	490	1895	1377	0.88
16	1790	2962	1817	968	463	1656	2121	223	0.98
17	1835	2685	1726	832	262	1965	1532	1163	0.90
18	1853	2401	2372	1204	352	1867	1783	168	0.98
19	1679	2982	2372	1021	175	2020	1271	480	0.96
20	1332	3800	1899	992	349	3182	446	0	1.00
21	1455	3089	2334	1064	433	1947	1230	448	0.96
22	2048	3425	2462	1079	316	1213	751	706	0.94
23	1957	2646	2289	803	395	1504	2148	258	0.97
24	1445	2691	2594	1013	427	628	1305	1897	0.84
25	1659	4266	2027	1115	559	628	1226	520	0.95
26	1477	2724	1769	931	424	2525	1787	363	0.96
27	1436	3004	1995	1156	343	1300	1904	812	0.93
28	1507	2685	2103	809	420	2517	1445	514	0.95
29	1955	2766	2451	934	426	744	1184	1540	0.87
30	1700	2082	2628	1053	400	2358	1148	631	0.94

Average CPU usage: 93%

Notes: (1) The data is given in thousands of bytes. (2) A time period represents 18 minutes during the nine-hour peak shift: 8.00 am to 5.00 pm.

2000000 bytes (from 12000000 to 14000000) significantly decreases the paging caused by insufficient memory, the resulting average CPU usage of 82% is still too high to be acceptable by management. Therefore, another simulation run was made increasing memory by another 2000000 bytes, bringing the total available main memory to 16000000. All memory requests by the workloads under this environment are now met and no paging, caused by insufficient resources, takes place. A decrease in average CPU usage to 72% overshot management's objective however, thus necessitating a final simulation run with 15000000 bytes of available CPU storage. Increasing the main memory resource from 12000000 to 15000000 bytes resulted in an average CPU usage of 76% (compared to 77% in the current system) with no paging caused by conflicts in CPU resource demands. These are acceptable results.

6. SUMMARY AND CONCLUSIONS

Since acceptable levels of productivity are currently being maintained, data-processing management is interested in what adverse affects the addition of CICS will have on the performance of the existing system.

This study was conducted for the following reasons: (1) to provide management with information on the workloads at current system operations, (2) to simulate the impact of adding a CICS to system operations workloads, and (3) to identify hardware modification alternatives which will compensate for the additional workload.

Management has two capital budgeting alternatives. The first alternative is to increase the main memory by 3000000 bytes (at a cost of $35000 per 1000000 bytes). This approach would reduce CPU usage to an acceptable level of 76%. This would not, however, allow for future computer growth. The second alternative available to management is to increase the main memory by 4000000 bytes. This approach will reduce the average CPU usage to 72%, and allow a substantial margin for computer system growth.

The simulation described in this paper fulfils the stated objectives, and provides a vehicle by which management may plan future growth/capacity requirements regardless of workload modifications.

REFERENCES

1. H. Artis and R. Paulhamus, Workload classification: problems and techniques. In *Symposium on Simulation of Computer Systems* edited H. Highland, pp. 95 SIGISM, New York (1976).
2. A. Esposito, A. Mazzeno and C. Savy, Systems performance evaluation: a simulation model for batch processing, in *Simulation of Systems 1979*, edited L. Dekker, G. Savastano and G. Vanteenkiste, pp. 1139–1148. North-Holland, New York (1980).
3. D. Ferrarri, Workload characterization and selecting in computer performance measurement. *Computer* **5**, 18–24 (1972).
4. G. Gordon, *Systems Simulation*, pp. 158–159. Prentice-Hall, Englewood Cliffs, New Jersey (1978).
5. A. C. Rucks, The FORTRAN synthetic program: a tool for simulating a computer workload. *Simulation* **36** (3), 73–81 (1981).
6. R. E. Shannon, *System Simulation: The Art and Science*. Prentice-Hall, Englewood Cliffs, New Jersey (1975).