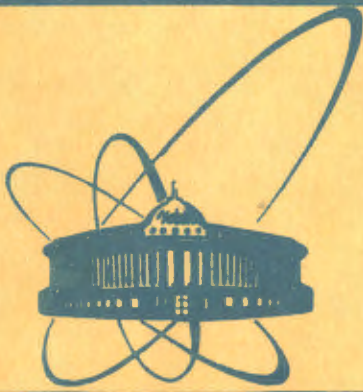


12/24-83



сообщения
объединенного
института
ядерных
исследований
дубна

6509/83

E11-83-522

Z.K.Loboz, T.I.Zaboy

**MEASUREMENTS AND TUNING
OF OPERATING SYSTEM
PERFORMANCE NOS/BE ON CDC-6500
AT JINR, DUBNA**

1983

1. TOOLS USED TO MEASURE AND ANALYSE SYSTEM PERFORMANCE

System work was analyzed by program package system performance analyzer (further referred as SPA) designed and written by Zdzislaw Loboż. This paragraph describes main characteristics of system architecture and program package used to measure system work.

1.1. System Architecture

Detailed system architecture is described in appropriate manuals. The parts more important from the point of view of our measurements are:

1. Two central processors (CPU'S). Because of their identical speed and management by system monitor they are considered as one processor.
2. Central memory (CM). Size 128K words.
3. Ten peripheral processors (PPU'S).
4. Six 100MB disks. They are connected to the system through one disk channel. Operating system resides on one of them. Three are for permanent files and Input/Output queues. Two are for user's private packs. Local (scratch) files may reside only on system disk or on permanent file disks.
5. Six 30MB disks. Used exclusively for users private packs.

1.2. SPA

Package consists of two main programs - one for gathering data about system work and one for analysis of this data. There are also other programs to assure proper work of main programs between deadstarts and ease analysis of received data.

1.2.1. Data Gathering - SPA (PP)

Data is gathered by a bouncing peripheral processor program - SPA(PP). It scans central memory in specified time interval and writes information obtained on each scan onto system file. Information contains actual values of about 400 system variables like, for example, IDLE CPU time, number of jobs in central memory, channel statuses and so on.

1.2.2. Data Analysis - SPACP

File produced by SPA(PP) is analysed by fortran program. On input to it we specify time period which we want to analyze, variables to be analyzed over specified time period and type

of analysis. For each value we can obtain following characteristics: average, minimum, maximum, standard deviation, histogram distribution, correlation with other values. In addition we have a possibility of combining several values written by SPA(PP) into one value analysed by SPACP. Due to package modularity we can easily analyze behaviour of system variables using other methods. (See App.B, App.C).

2. METHODS OF MEASURING AND ANALYSIS

Interval of central memory scan by SPA(PP) was 60 seconds 1440 times per day. It proved to be enough for statistical analysis of received data and system file produced by SPA(PP) was of reasonable size - about 280 PRU per day. After installation of SPA in the system the received data were analysed daily to find main system characteristics and estimate which system variables are most important - we can analyse about 400 of them, but only a small subset is critical to system performance.

These analyses have shown, that system bottlenecks are delays caused by jobs waiting for disks input/output operations and peripheral processors. Additional counters were installed in the system to obtain more detailed information about these parts of the system.

Initial measurements have shown that the system work may be divided into two parts - peak hours and normal hours - and the analysis of these two kinds of system work was performed separately: from 13 till 22 hours and from 22 till 8 hours. The remaining period - from 8 till 13 hours is an engineer's time.

3. RESULTS

This part describes values received for main system characteristics. Only typical values are given - system was measured for several months and printing all received values here would result in report of unreasonable size.

The word day denotes time interval from 13 to 22 hours - i.e., time which may be described as peak hours - due to the number of logged in Intercom users, jobs processed and so on. The word night denotes time interval from 22 to 8 hours in the morning. The average values below are for April-May 1983. These months were chosen for analysis, because at this time all additional counters were installed in system - and we could analyse system work in detail.

3.1. IDLE CPU Time

For measured period of time in analyzed time intervals (days and nights) IDLE CPU time varied from 0.1 to 0.5. Average value

for working days - Monday to Friday - was 0.34. During night hours IDLE CPU time was on average 0.23. On week-ends during the day IDLE CPU time is between 0.20 and 0.25, during the night it is sometimes less than 0.10. The average IDLE CPU time varies significantly from month to month, for example, in November 1982 it was about 0.40. Variations in IDLE CPU time measured by SPA are similar to those measured by dayfile analyser. The differences between months are about 10%. This differences are mainly due to job mix prevailing on a given month - CPU bound jobs, short I/O bound jobs and so on.

3.2. Peripheral Processors

IDLE PPU time - for all PPU'S - ranges from 2.5 to 4.5 seconds for one physical second. It is smaller during the day 2.5 to 3.0 - when a lot of I/O operations are performed. Another variable which also describes the use of peripheral processors is average number of free PP. For working days it ranges from 1.2 to 1.9 during day hours and from 2.3 to 3.1 during the night.

From analysis of this variable we can see that maximal number of unsatisfied PP calls (in other words queue to PP) is 10 on average day. In all, during the day, queue to PP exists for 6 to 16 percent of day time. The same percentage for nights ranges from 0 to 2%.

3.3. Disk Calls

In NOS/BE a program to perform disk Input/Output operation calls the system. If disk channel is busy servicing other call - the next call must wait. The number of disk calls per second ranges from 18 to 21 during the day and from 16 to 19 during the night - for all 100 MB disks. For all 30 MB disks number of calls per second varies from 0.5 to 1.5 - small disks are used for private packs only.

System variable, which proved to be important to system performance is a length of disk queue. Disk calls which cannot be satisfied immediately are waiting in disk queue - and programs issuing them cannot use CPU. Average disk queue length varies from 2.5 to 2.9 during the day and from 1.6 to 1.8 during the night.

Although there are 12 disks in system disk queue to each disk is not the same. There is no queue to 30 MB disks. There is practically no queue to two 100 MB used for private packs. For average queue length during the day - 2.7 in all - 1.5 comes from system disk, 0.5 from permanent file master disk, 0.3 from each of two remaining permanent files disks.

3.4. PP Program Calls

Each programs in NOS/BE to perform Input/Output operation on any device must call PP program. Number of calls per second for PP programs varies from 35 to 55.

There are about 500 PP programs in the system and they usually reside on disk; most frequently used ones reside in central memory. System, however, gives us no tools to determine exactly which programs are frequently called - and to determine how many calls per second are made for disk resident PP programs. After installation of appropriate counters it was found that number of calls per second during the day varies from 5.0 to 6.5 - i.e., about 30% of all disk calls are calls for disk resident PP programs. Calls for disk resident PP programs comprised on average 10 to 12% of all calls for PP programs. (See App.A).

4. IDLE CPU TIME - REASONS

Analysis of correlations between system variables has shown, that IDLE CPU time is mainly due to disk queue. Disk queue is also responsible for increasing PPU activity - because PP programs are waiting for information from disk. Analysis of plot "average disk queue length" versus "average IDLE CPU time" for 44 experimental points has shown, that we may describe IDLE CPU time as a linear function of disk queue length:

$$\text{IDLE CPU} = 0.07 + 0.11 * \text{disk queue.}$$

The correlation coefficient between the above function and results of experiments is 0.90.

In actual system version the dependence between disk queue length and number of disk calls per second (for average values) is

$$\text{disk queue} = 0.28 * \exp(0.12 * \text{disk calls}).$$

Correlation coefficient between the function and results of experiment is 0.97.

Dependencies described above are statistical. For disk queue length 2.5 measured IDLE CPU time varied between 0.25 and 0.35 - because IDLE CPU time is produced by many factors - from untypical job mix to wrong operator intervention. What's more we have to deal with working system? At any time a new job may be read into the system or end because of error, at any time a user may LOG any terminal IN or OUT. Therefore such variations should be expected - but big correlation coefficients show, that there are certain statistical regularities in system work.

Although from the above functional dependencies we can compute, that for 19 disk calls per second IDLE CPU time should

be 0.35 this is only an average (expected) value of IDLE CPU time.

From this functions we can see, for example, that by reducing number of disk calls per second from 19 to 15 we can reduce IDLE CPU time from 0.35 to 0.25 - i.e., by 10%.

5. EXPERIMENTS

After finding regularities and main bottlenecks in system work some experiments were made to assess possibilities of improving operating system efficiency. The objective was to curtail IDLE CPU time without worsening any other system characteristic. IDLE CPU time, as mentioned earlier, depends on many factors - so to estimate improvements in average IDLE CPU time after changing - for example number of disk calls we would have need at least several days of measurements. There was not enough time available for such measurements, therefore we were forced to use functional dependencies described earlier and look rather on disk queue than on IDLE CPU time - mainly because disk queue length and distribution is much more stable than IDLE CPU time.

5.1. Calls for Disk-Resident PP Programs

The total size of Central Memory Resident PP programs was 34000B words. The number of calls per second for Disk-Resident PP programs was on average 5.5. Although this comprised 30% of all disk calls - it was about 12% of all calls for PP programs. The CM-Resident subset of PP programs was determined by CDC recommendations and observation of system work.

Changing the system to count the number of times each individual PP program is called enabled us to improve this situation and find optimal subset of CM Resident PP programs for each given CM Resident size - and predict a number of disk calls for each subset.

Two experimental subsets of CM Resident PP programs were tested. The first one to minimize number of calls for Disk Resident PP programs with maximum size of CM Resident 35000B words. The second one to minimize size of CM Resident with the same number of disk calls per second as for "standard" resident-5.5. The results were as follows

SIZE	DISK PP CALLS	DISK QUEUE	SYSTEM DISK	PERM. A	FILE B	DISKS C
34KWB	5.5	2.7	1.5	0.5	0.3	0.3
35KWB	1.5	1.9	0.7	0.5	0.4	0.3
20KWB	5.0	2.5	1.4	0.5	0.3	0.3

For Resident minimizing the number of disk accesses the reduction in number of disk calls was 4 per second. The total queue in result was shortened by 0.8. As PP programs reside on system disk the queue length for permanent file disks was not affected but the queue length to system disk was halved. Additional consequence of using such a resident was decrease in PPU activity by about 10% to 20%. This was due to the reduction of "busy waiting" state of PPU - when PP program waits for information from disk occupying PP - but doing nothing in fact.

From the results of experiment with a resident minimizing central memory occupied by PP programs we can see, that keeping the number of disk calls per second (and disk queue) on the same level as before the experiment we can obtain 14000B words of central memory more. The number of disk calls during the night is smaller - so the gain during the night hours may be even 20000B words - and outside Intercom hours even 24000B words.

5.2. Changing Distribution of Disk Queues

From the previous experiment we have seen, that it is possible to reduce length of disk queue by reducing a number of disk calls. There are, however, other possibilities - changing distribution of disk calls between disks. The total number of disk calls per second is about 19 - but the queue lengths for each individual disk differ greatly. This suggest that the number of calls for each disk is not identical. The system does not count disk calls for each disk - so we must use a simple model to estimate what gains can be expected from changing distribution of disk calls.

5.2.1. Model of Disk Calls Distribution

According to queue theory the average queue length for a given system is given by

$$\text{QUEUE} = \text{RHO} \cdot 2 / (1 - \text{RHO}).$$

Where $\text{RHO} = \text{RQS} / \text{SPEED}$; RQS is a number of requests per time unit, speed is disk speed measured by a number of requests that may be satisfied in time unit.

This function is highly unlinear and for systems with RHO greater than 0.5 we can expect a significant reduction in queue length for small changes in RHO. It means, that for systems working at their limits even small reductions in number of request per second should result in significant reductions of queue length. The above is, of course, true also for reduction of total number of disk calls described in experiment with PP

programs. Let's see how it should work for different distributions of calls for individual disks.

Suppose, that total number of disk calls per second is 19 and speed of each disk is 14. With distribution of calls between four disks like (5,5,5,4), the total queue length will be 0.71 (0.2, 0.2, 0.2, 0.1). If we change the distribution to (10,3,3,3) the total queue length will rise to 1.96 (1.8, 0.06, 0.06, 0.06). Now, suppose that the system will try to change a placement of only one call - from disk 4 to disk 1 - without changing total number of calls. For distribution (11,3,3,2) we have total queue length 3.06 (2.9, 0.06, 0.06, 0.02) - reassigning one disk request has changed queue length by 30%.

The above is only a simplified model. In our system disk speed is not constant - it changes with number of requests per time unit. Nevertheless it is obvious, that changing distribution of disk calls for individual disks we can change length of disk queue.

5.2.2. Second Disk Channel

In this installation there are two kinds of disks 100MB and 30MB. Total number of request per second for 30MB disks is on the average 0.5 to 1.5. This is very small in comparison with 100MB disks with their 19 calls per second - but small disks are used for user private disks, while on 100MB disks we have local (scratch) files, system and permanent files.

To reduce number of calls for 100MB disk and distribute disk calls between greater number of disks one 30MB disk was added to the system and defined as scratch. The total number of calls remained on the same level - 19, but not 5 of them were for additional disk. The distribution of queue lengths to 100MB disks was changed from previous (1.5, 0.5, 0.3, 0.3) to (0.8, 0.2, 0.1, 0.0) - total queue to 100MB disks was reduces from 2.7 to 1.1. But the request for 30MB disk created - non-existing before - queue to small disk with length 1.9. So, the total queue to all disks was 3.0. Small disks are slower and cannot handle such a number of disk calls per second as 100MB disks.

In the next stage of the experiment two small disks were added to the system for scratch files. Total number of disks requests remained on the same level, but now 6 of them were for two additional 30MB disks. Distribution of queues for 100MB disks was (0.7, 0.2, 0.1, 0.1) - total queue to 100MB disks was 1.1. Queue length for two additional disks was 1.2, - so total queue length for all disks was 2.3.

The above results show, that the number of requests transferred to each added 30MB disk shouldn't be greater than 2 per second - to avoid queue to small disks. To produce such a num-

ber of calls for each small disk added to system we should change relative weights of 100MB and 30MB channels.

Taking into account difference between number of disk calls per second and IDLE CPU time we can expect decrease in IDLE CPU time by 5% for one small disk added to the system and by 10% for two small disks added to the system - if the number of requests to these disks will be on the level 2 per second for one disk.

5.2.3. Uniform Distribution of Calls for 4 100MB Disks

In the present system configuration we have 6 100MB disks, but two of them are used exclusively for user's private packs. From the remaining 4 - one contains system files; the other 3, permanent files. Local (scratch) files may reside on any of these four disks. There are big differences in queue lengths to these disks - the distribution is (1.7, 0.5, 0.3, 0.3). The longest queue is to system disk - but this is due not only to the calls for disk resident PP programs or other system programs. The problem is with local files. When the user program (or Intercom) needs a local file, system finds a disk with a biggest amount of free space. And there is more free space on system disk than on any other permanent file disk - most local files are assigned to system disk. So, from 19 calls per second to all 100MB disks there are at least 10 to system disk - 5 for disk resident PP program and about 5 for local files.

The simplest method to check if it is really so is to catalog a big dummy file on system disk and, by this, force the system to use other disks for local files more frequently. After such an experiment distribution of queue lengths changed from (1.7, 0.5, 0.3, 0.3) to (1.1, 0.5, 0.4, 0.2) - and total queue length was reduced from 2.7 to 2.2.

This result means, that by manipulations with an assignment the local files and number of calls for disk resident PP programs we can significantly reduce queue length.

6. CONCLUSIONS

The results of measurements and data received after experiments show, that it is possible to increase CPU efficiency by up to 15%. It is also possible to increase available memory by up to 20000B words.

Increase in CPU utilization may be obtained by reducing the number of disk calls and/or even distribution of calls between available disks for scratch or frequently used system files. Additional memory may be obtained by optimizing contents of central memory resident.

The above may be obtained in a number of ways - the most efficient ones require certain changes in the system. Some of them do not - but they may be inconvenient to the computer center because of user demands - like using three 30MB disks for scratch files.

To decide which way will be chosen is not an analyst work. In our opinion a system analyst should only provide a list of possibilities and expected results. Choosing between such a possibilities involves also a policy of computer center towards its users and should be solved on managerial level.

As a typical example of such decision lets consider to CDC-6500 connection of additional terminals. If new terminals will be used as frequently as the old ones, we can expect increase of IDLE CPU time by up to 15% - due to the additional disk calls. But if we need these terminals more than 15% of processor time, there is no reason not to connect them. The same holds for other similar trade-offs: Intercom response time versus IDLE CPU time, and of these two versus number of terminals connected and so on.

The results of measurements proved, that for a given situation we can optimize usage of computer resources - but they cannot give answer to a question what situation we should choose.

ACKNOWLEDGEMENTS

We are very grateful to Prof. N.M. Gvozden, Prof. I. Kolendowski and Dr. V.P. Shirikov for suggesting possibility of realizing our task and also thank them for support and encouragement. And we are very grateful to Dr. J.I. Shelonzsev for his help which succeeded us to do this work in such a short time.

REFERENCES

1. NOS/BE1 System Programmer's Reference Manual (60494100), Sunnyvale, California, USA.
2. Ferrari D. Computer Systems Performance Evaluation. Prentice-Hall, Englewood Cliffs, New Jersey, 1978.
3. WenderMerr T. GSS-A Package for Performance Analysis. RUG, Proceedings, Ecodu 24/VIM27, Montreux, Switzerland, September 26-29, 1977.
4. Вентцель Е.С. Исследование операций. "Сов.радио", М., 1972.

FREQUENCY ANALYSIS OF PP PROGRAM CALLS

REC TUE 05/09/83 16.28.27
REC TUE 05/07/83 19.16.27

TIME 136805 (3M 2800S) ALL CALLS 669062 (48.91 CALLS/SEC) DISK CALLS/SEC 5.41 (11.07%) CHR SIZE 212668 (8886D)

ITEMS PRINTED

NAM -- PP PROGRAM NAME
%CUM -- CUMULATIVE PERCENTAGE OF ALL CALLS
LENZ -- PROGRAM LENGTH IN % OF ACTUAL PPGCHR SIZE
CLEND -- CUMULATIVE LENGTH (DECIMAL)
RS -- RESIDENCY BITS - O4 = C11,00=DISK
LENG -- PROGRAM LENGTH (OCTAL)
CLEM -- CUMULATIVE LENGTH (DECIMAL)
DCALLS -- DISK CALLS PER SEC 3M EFFECT
EFF -- CALLS/LENGTH

	NAM	RS	%ALL	%CUM	LENG	LEND	LENZ	CLENZ	CLEM	CLEND	DCALLS	EFF
1	CIO	04	27.90	27.90	0425	277	3.12	3.12	00425	277	35.26	492.647
2	4ES	04	19.06	46.97	0320	208	2.34	5.46	00745	489	25.94	4481976
3	1ES	04	2.58	49.54	0072	58	.65	6.11	04037	543	24.68	2173951
4	HSG	04	3.42	52.97	0126	86	.97	7.08	01165	629	23.00	1945804
5	4DO	04	1.95	54.91	0147	103	1.16	8.24	01334	732	22.05	925026
6	1CI	04	3.91	58.82	0415	269	3.03	11.26	04751	1001	20.14	710477
7	1IT	04	6.45	65.27	1243	675	7.60	18.86	03214	1676	16.99	467099
8	1QP	04	4.07	69.34	1062	562	6.32	25.19	04276	2238	15.00	354102
9	2QS	04	3.91	73.24	1053	555	6.25	31.43	06351	2793	13.09	344357
10	1RN	04	2.03	75.28	0555	365	4.11	35.54	06126	3458	12.09	272270
11	2TW	04	.49	75.77	0146	102	1.15	36.69	08274	3260	11.85	235208
12	1BR	04	2.01	77.78	0647	423	4.76	41.45	07143	3683	10.87	232362
13	1CL	04	.85	78.63	0272	186	2.09	43.54	07435	3869	10.45	224407
14	2IR	04	.46	79.09	0157	111	1.25	44.79	07614	3980	10.22	204875
15	LDL	04	.84	80.04	0346	230	2.59	47.38	10162	4210	9.76	200387
16	3TT	04	1.12	81.16	0436	286	3.22	50.60	10620	4496	9.22	191443
17	1QM	04	.47	81.63	0056	46	.52	51.11	10676	4542	9.11	181000
18	LDV	04	.70	82.02	0277	191	2.15	53.26	11175	4733	8.79	178693
19	2CC	04	1.80	83.83	0735	477	9.37	58.63	12132	5210	8.01	164450
20	3TZ	04	.75	84.57	0364	244	2.75	61.38	12516	5454	7.64	149524

APPENDIX A

SYSTEM PERFORMANCE REPORT

FROM FRI 17/06/83 21 H 59 M TO SAT 18/06/83 7 H 59 M SAMPLES 595

VALUE NAME	AVERAGE	MIN	MAX	STD	DISTRIBUTION									
					0	1	2	3	4	5	6	7	8	9
CPU IDLE	.20346	0.000	.818	.154	.272	.341	.207	.111	.060	.030	.015	.012	.002	0.000
SCHSHV TOT	.02802	.003	.102	.021	.459	.291	.129	.092	.025	.003	0.000	0.000	0.000	0.000
IDLE PP	4.06869	.825	5.720	1.079	0.000	.003	.008	.039	.045	.066	.121	.185	.153	.380
PP FREE	2.92101	-5.000	5.000	1.746	.008	0.000	.000	.010	.015	.042	.072	.163	.250	.430
PP CALLS	40.36255	12.031	233.188	28.758	.466	.185	.094	.052	.040	.034	.024	.007	.013	.086
DPP CALLS	3.69480	1.719	9.828	1.430	0.000	.082	.358	.224	.141	.084	.047	.020	.024	.020
DAT 2 CNT	.46011	0.000	9.609	1.267	.684	.066	.029	.017	.042	.018	.005	.012	.012	.146
DAT 3 CNT	13.73272	.031	36.641	7.156	.618	.119	.136	.045	.047	.025	.005	.003	0.000	0.000
RQS QUEUE	1.46050	0.000	8.000	1.586	.570	.247	.141	.113	.076	.042	.005	.003	.003	0.000
RQS6 QUEUE	.33950	0.000	4.000	.685	.760	.163	.057	.018	.002	0.000	0.000	0.000	0.000	0.000
RQS7 QUEUE	.09748	0.000	2.000	.308	.906	.091	.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RQS8 QUEUE	.42773	0.000	3.000	.390	.891	.092	.015	.002	0.000	0.000	0.000	0.000	0.000	0.000
RQS 9 QUEUE	.00840	0.000	1.000	.091	.992	.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
RQS10QUEUE	.80168	0.000	6.000	1.077	.518	.291	.109	.044	.034	.003	.002	0.000	0.000	0.000
OCCPD CPT	5.86050	1.000	10.000	2.117	0.000	.002	.094	.045	.116	.168	.182	.155	.118	.121
INT CRDS	.01988	0.000	.172	.028	.524	.267	.136	.050	.018	.002	.002	0.000	0.000	0.000
ACTV I USR	.54454	0.000	5.000	.786	.593	.306	.072	.022	.005	.002	0.000	0.000	0.000	0.000
SWAP TIME	.15105	0.000	.644	.109	.210	.309	.175	.133	.076	.054	.022	.017	.003	.002
NO OF SWPS	.17503	0.000	.703	.132	.239	.287	.187	.121	.079	.049	.025	.007	.005	.002
WDS SWPD	.75960	0.000	2.219	.451	1.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ALL QUIET	.01129	0.000	.119	.018	.914	.015	.007	.008	.015	.010	.002	.002	.003	.024

APPENDIX B

SUMMARY STATISTICS

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1 CPU IDLE	0	50	-59	-34	39	52	11	60	45	25	5	21	2	36	15	23	17	38	31	30	38
2 SCHSMV TOT	30	0	-53	-34	18	56	8	49	37	23	16	17	16	26	33	57	47	79	81	68	62
3 IDLE PP	-59	-53	0	66	-47	-66	-28	-76	-58	-33	-10	-25	-9	-45	-42	-45	-36	-47	-53	-28	-60
4 PP FREE	-34	-34	66	0	-32	-45	-15	-46	-66	-50	-11	-12	0	-68	-35	-36	-34	-33	-39	-20	-38
5 PP CALLS	39	18	-47	-32	0	23	12	65	39	28	3	14	-2	32	29	8	13	12	15	9	13
6 DPP CALLS	52	56	-66	-45	23	0	7	64	47	20	9	15	15	42	39	56	37	54	61	35	55
7 DAT # CNT	11	8	-25	-15	12	7	0	12	12	6	0	2	-3	6	10	10	9	6	6	5	7
8 DAT 3 CNT	60	49	-76	-46	65	64	12	0	55	36	10	24	7	42	47	32	40	43	26	43	43
9 RRS QUEUE	45	37	-58	-66	39	47	12	55	0	60	26	38	2	81	43	29	35	32	36	19	38
10 RQ86 QUEUE	25	23	-33	-30	28	20	6	36	60	0	2	15	-2	18	25	12	22	16	17	7	21
11 RQ57 QUEUE	5	16	-10	-11	3	9	0	10	26	2	0	-2	-3	10	3	7	9	13	13	11	10
12 RQ58 QUEUE	21	17	-25	-12	14	15	2	24	38	15	-2	0	2	8	13	9	14	12	13	8	14
13 RQ5 0 QUEUE	2	16	-9	0	-2	15	-3	7	2	-2	-3	2	0	-3	7	13	8	16	16	11	17
14 RQ510 QUEUE	36	26	-45	-68	32	42	6	42	81	18	10	8	-3	0	38	28	28	31	17	30	30
15 OCCPD CPT	15	33	-42	-35	29	39	10	47	43	25	3	13	7	38	0	24	37	31	35	21	33
16 INT CMDS	23	57	-45	-36	8	56	10	32	29	12	7	9	13	28	24	0	32	63	72	46	46
17 ACTV I USR	17	47	-36	-34	13	37	9	32	35	22	9	14	8	28	37	32	0	54	59	45	37
18 SWAP TIME	38	79	-47	-33	12	54	6	40	32	16	13	12	16	26	31	63	54	0	95	91	66
19 NO OF SWPS	31	81	-53	-39	15	61	6	43	36	17	13	13	18	31	35	72	59	95	0	84	67
20 NOS SWPS	30	69	-28	-20	9	35	5	26	19	7	11	8	11	17	21	46	45	91	84	0	47
21 ALL QUIET	38	62	-60	-38	13	55	7	43	38	21	10	14	17	30	33	46	37	66	67	47	0

Received by Publishing Department
on July 20, 1983.

Лобоз З.К., Забой Т.И. E11-83-522
Измерения и оптимизация операционной системы на CDC-6500, ОИЯИ, Дубна

Исследования работы операционной системы показали, что основными источниками простоя центральных процессоров являются дисковые запросы и насыщение периферийных процессоров. Наибольший вред дает насыщение дискового канала, т.к. это приводит к насыщению и периферийных процессоров. На CDC-6500 были проведены различные эксперименты, чтобы найти методы улучшения коэффициента использования центральных процессоров без ухудшения других характеристик системы. Результаты экспериментов показали, что коэффициент использования центральных процессоров может быть улучшен на 15% в дневное время за счет более равномерного распределения запросов между системными дисками, усиления использования других дисков и контроля числа дисковых запросов от системы. Существует возможность высвобождения дополнительно от 14000В до 24000В слов центральной памяти за счет выбора оптимального набора системных программ резидента центральной памяти.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна 1983

Loboz Z.K., Zabay T.I. E11-83-522
Measurements and Tuning of Operating System Performance NOS/BE
on CDC-6500 at JINR Dubna

Investigation of operating system work has shown, that main sources of central processor idle time are disk requests and peripheral processors saturation. Of these two disks channel saturation is more damaging because it produces PPU saturation. To find a method for improving CPU utilization factor without worsening other system characteristics several experiments were made. The results of experiments show that CPU utilization coefficient may be improved by up to 15% during daytime by evenly distributing disk requests between available disks, more active usage of other disks and controlling a number of disk calls for system programs. There exists also a possibility of obtaining additional 14000B to 24000B words of central memory by selection of appropriate set of central memory resident system programs.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna 1983