Aug 13 1971

A NOTE ON CHARGING

INTRODUCTION

Within the next few weeks it will be possible to begin cnarging for the use of TSS. In fact we will probably begin testing the charging ~~algorithm~~ procedures on live customers within a week. This note will explore various rationals e for setting rates and thc ~~[illegible]~~ resulting charges for a few standard tasks.

RATIONALS E

There are a number of different ways to set rates. The method used in this note is to set a nominal rate for the machine as a whole. This rate is then partitioned among the various resources ~~for~~ which we can account. The actual rate for the machine as a whole will then depend on the percentage of utilization of each of these accountable resources. The nominal rate for the machine as a whole is then adjusted on an ad hoc basis to obtain the desired income.

Another method, not used in this ~~XXXX~~ note, is to attempt to set the rate for each piece of ~~X~~ equipment so as to pay ~~[illegible]~~ its costs. ~~Thus if~~ Thus, for example, if disk storage became full ~~XX~~ there would be enough income from the disk itself to pay for another. This method has a number of problems~~X~~. ~~[illegible]~~ Not all components can be extended at will and ~~not~~ allcomponents are not accounted. Moreover, this method makes no allowance for various overhead costs, such as system programmers. It might be worthwhile to examine this method later.

Once a nominal overall rate has been set it must be partioned among the accountable components. ~~[illegible]~~ The obvious method that presents itself is to devide the rate porportional to the cost of each component. This leads to similar problems to the other method, i.e. not all components can be accounted, and not all of each accounted component can be accounted. Another problem is that one of the accountable ~~[illegible]~~ items, connect time, is not a component.

We can account for parts of all the major components; cpu, ecs and the disk. One procedure is to just ignore all other components, and that is what we do in this note. Another question is whether to ~~wait~~ weight each component by its full cost and require tha accountable portion to pay for the full weight, or to weight each component by the cost of its accountable portion. This is a real problem for ECS since we can only account for half of it, Finally there is the question of how to determine the cost for a component, by the original cast to the computer center, or by a probable replacement cost. These last 2 questions lead to 4 possibilities, and in the later computations we examine all of them.

ac

its

STANDARD TASKS

There are a number of standard tasks considered in this note. The first is the null task. ~~What~~ The cost ~~would be~~ for just loggink in to the systøm and doing nothimg. The so called connect charge. This must be distinguished from a charge for connect time alone, since while a tty is logged in it is sitting on some resources that are accounted. The total connect charge would be the sum of the charge for connect time alone and the charge for the resources researved to the tty.

? It will turn out later, ~~that~~ under the charging assumptions made in this note, that the charge for the resources researved to a ~~ttx~~ logged in tty are quite high. This is due to a number of inefficiencxies in the current system. It is hoped that they will be reduced in the next 6 months. In the meantime we propose a negative charge for connect time, so as to reduce the connect charge to a reasonable amount.

(A)→ The next tasks considered will ~~bx~~ involve the use of the Scope Simulator. ~~XXXXXXXXXX~~ All estimates of charging ~~wixxix~~ assume that ecs will be used for the same length of time as the cpu is used. This would be apporximately true if only the one user is on the system. Since the ecs costs will be low compared with the cpu costs, the errors can be ignored under low load. Under high load the increase in cost for ecs will become significant, and this may lead users to avoid the system during high load.

All of these scope tasks will assume a filed sp length of 45K (octal) , sufficient for large assemblies. The scope simulator requires 100K (octal) of ~~x~~ swapped ecs under these conditions.

(B)→ The particular scope tasks will be; null, null assembly, and large assembly. Under the current system the null use of the scope simulator, call and return, takes abøut 12 seconds. Running a null Nompass assembly ~~xx~~ raises the total to about 20 seconds. ~~Kixxxixxkx Kixxxixx~~

The large assembly ~~is one of the decks used in~~ used in the examples is a component of the system itself. The cost on the A machine for assembling the deck under Mompass is 1 minute of cpu time and 80 seconds of ppu time. At $400 per hour on the A machine the charge would be about $6.70 . Under our current system the cost seems to be about 3 minutes, including the overhead for ealling the scope simulator.

(A) The total resources used by a logged in tty are 7.5K (decimal) ecs and ~~XXXXXXXXXXXXXXX~~ ~~128~~ 32K ~~(decimal) of~~ some disk space. The ECS space comes from 2 types of overhead. The first is system overhead of ~~fixed ecs space of~~ 4K ~~X~~ of fixed ecs space. We expect this to reduce over the next 6 months. The second is due to a crude algorithm for the control of swapped ecs space. We expect this to reduce to near zero under the forced swapped procedures to be installed late this year. The cost of the disk space is ignored, since it is about $.10 at most.

~~We will consider 2 kinds of users, small and large. The small users will have about 10K disk space, about 150 sectors. The large users will have about 32K, about 500 sectors.~~

(B) The scope simulator also makes use of the disk. In order to run large assemblies more disk space than the nominal amount for a logged in tty must be researved. Compared to the cost for cpu this will be small, on the order of ~~10 dollars~~ ~~a dollar~~ 10 dollars an hour, and we will neglect it.

The final task to be considered is that of permanent file storage. We will attempt to compute the costs on a per month basis. Since the system accumulates chagges for disk space only while the system is actually running, we need an estimate of the number of hours the wywtem will be up in a month. A month has about 700 hours, there fore it will be unlikely that the system will be up more than 350 hours.

Next we need estimates of the amount of permanent disk space needed. We give these for three classes of users: small, medium and large. The small user is ~~considered~~ a student with a very small program to save, say about 1 or 2 pages. We estimate that about 800 words or 12 sectors will be sufficient. The medium user has maybe a 50 page program, about 20 thousand words or 300 sectors. Finally, the large user has maybe 120 thousand words, or about 1800 sectors. (We can not support too many large users.)

RATE SETTING

In order to set a rate by the method we have choosen, we ~~need~~ a nominal rate for the whole machine. For this purpose we have choosen $400 per hour. This figure must be compared with the A machine, which is slighlty in excess of $400 an hour. The B machine is ~~probably~~ somewhat cheaper than the A machine since it does not have as much central memory, nor as many printers or tape drives. The operator overhaed for running TSS is considerably less than the Scope system, at present requireing an ope rator only at start up and shutdown, plus an ocasional tape mount. The programming staff for TSS is also somewhat less than for the A machine.

We expect the TSS to give a much higher rate of utilization for the accounted components than does the scope system, thus inherently producing a higher rate of income for a fully loaded system.

contradiction?

In view of the results that follow, we sould suggest starting with a basic rate of $300 per hour if the desire is to produce a competitvely priced system.

Having choosen a basic rate, we need the comparative costs of the various components in order to divide up the rate. The following table contains the basic information used in the subsequent calculations. These values were obtained from Ken Hebert on Aug 12, 1971.

component	original cost	probable replacement cost	standard unit	total standard units (NSU)	accountable units (NASU)
cpu + 32K cm	$.69*10^6$	$.5*10^6$	1 hour	1	.9
300K ECS	$.48*10^6$	$.6*10^6$	1 K-words-hr	300	160
1/2 disk	$.26*10^6$	$.10*10^6$	1 K-sector-hr	1020	765
other	$.08*1o^6$	$.10*10^6$	--	--	--

$400 per hour, 24 hours per day, 365 days per year amounts to $\$3.5*10^6$.

The following tables contain the computation of the rates per standard unit under the different assumptions. First we define the symbols used:

R = total rate to be distributed (we use $400 per hour.)
cc = component cost
tc = total cost for all components
nsu = number of ~~xxxxxxxxxx~~ standard units in the component
nasu = number of accountable standard units in the component
ac = cost of accountable portion of a component = cc X nasu / nsu
tac = total cost of accountable portions of components

Next we give the formulas for the two methods:

method A

rate / s.u. = R X (cc / tc) X (1 / nasu)

method B

rate / s.u. = R X (ac / tac) X (1 / nasu)

Finally we give the tables containing the computations:

method A

		(original costs)			(replacement$costs)	
component	cc	cc/tc	rate/su	cc	cc/tc	rate/su
cpu + 32K cm	.69	.46	$205.	.5	.38	$170.
300K ECS	.48	.32	$.80	.6	.46	$1.15
1/2 disk	.26	.17	$.089	.1	.08	$.042
other	.08	.05		.1	.08	
tc =	1.51			1.3		

method B

	ac	ac tac		ac	ac tac	
component	ac	ac/tac	rate/su	ac	ac/tac	rate/su
cpu + 32K	.62	.59	$263.	.45	.53	$236.
300K ECS	.25	.23	$.58	.32	.38	$.95
1/2 disk	.19	.18	$.094	.08	.09	$.047
	1.06			.85		

COSTS PER TASK
~~costs per task~~

The following table gives the charges for the tasks described above in the 4 cases of rate division considered.

~~method A method A method B method B~~
~~original replacement~~

	original costs		replacement costs	
	method A	method B	method A	method B
connect time				
~~per hour~~ on	~~$5.24~~	~~$4.05~~	~~$10.50~~	~~$6.75~~
XXXXXXXXXXXXXXX				
XXXXXXXXXXXXXXXXXXXXXXXXXX				
scope simulator				
per hour for cpu and ecs	$230.	XXXX. $281.	$207.	$266.
null call	$.77	$.94	$.69	$.89
null Nompass assembly	$1.28	$1.56	$1.15	$1.48
large Nompass assembly $6.70 an A mch	$11.50	$14.00	$10.04	$13.30
Disk storage/month				
(at 350 hours/month)				
small (800 wds)	$.37	$.40	$.18	$.20
medium (20K wds)	$9.40	$9.90	$4.40	$5.00
large (120K wds)	$56.	$59.	$26.	$30.

~~Connect time - per hour~~		~~$5.24~~	~~$4.05~~	~~$10.50~~	~~$6.75~~
	~~ecs cost~~				
~~(150 sec)~~	~~small disk~~	~~.01~~	~~.01~~	~~—~~	~~—~~
	~~total~~	~~5.25~~	~~4.06~~		
~~(500 sec)~~	~~large disk~~	~~.45~~	~~.29~~	~~—~~	~~—~~
	~~total~~	~~5.70~~	~~4.34~~		

FINAL.

On the basis of the above discussion, I propose the following:

The basic charge rate be $300 per hour

Method B, based on xxpxx original osts be used to calculate tx the rates.

A negative connect charge be used so that the basic charge for connect time is about 1 or 2 dollars an hour.

TXXXXXXXXX

This results in the following rates:

cpu	$200 per hour
ecs	$0.43 per K-hour
disk	$0.07 per K sector hour