BILLING FOR COMPUTER SERVICES IN A UNIVERSITY ENVIRONMENT

by

Wallace E. Killcreas

Introduction

Billing for the services of a large scale computing system is a rather difficult task, which requires a great deal of attention at both administrative and technical levels. When the computing system is administered by a university, the billing process becomes more complex.

Complexity is introduced by the nature of the beast itself. Charges must be levied for direct computer usage. These charge areas include usage of the central processor and its associated peripheral devices as well as materials (cards, paper, magnetic tape, etc.) used. There are charges associated with user services such as programming and key punching. Even charges for materials become complicated on a large system if performance data is automatically collected. For example, if a user generates 500 pages of print, examines a summary and decides not to print the 500 pages, he could get billed for paper that wasn't used. On the contrary, if the same user physically printed the 500 page report 10 times, he could use 5000 pages, and unless careful management were exercised only pay for 500 pages. Programming services are a tougher problem. If a program is to be used over and over on a production basis, it will generate income for a computing center over a long period of time. Programming costs cannot always be justified simply on the basis of the professional time required to accomplish the programming task. In some cases, all programming costs must be collected upon completion of the programming assignment. In other cases, the direct programming costs can be discounted, and regarded as an investment.

Charges directly related to the central processor can be very complex indeed. For example, should one hour of CPU (central processing unit) time cost the same as one hour of I/O (input/output devices such as disks, drums, tapes) time? Also, should higher priority runs pay relatively more? Finally, should a run using 60K of core storage pay the same rate for CPU time as a run using only 15K? These are questions that must be addressed at both administrative and technical levels.

The diversity of user groups encountered in a university computing environment is an additional complicating factor. University user groups which might require different charge structures include educational, administrative, sponsored research, and unsponsored research. Even non-university user groups sometimes require different charge structures. These include private industry, government agencies, and government sponsored agencies under the umbrella of the university.

Administrative attention by the computing center director is the first step toward the development of a billing algorithm that will respond to the financial requirements imposed on the computing center by upper management as well as charge fairly for services rendered in a manner that is easily understood. The director must recognize that a computer system is a limited resource, and that billing is an effective mechanism to aid him in efficiently managing it.

Technical expertise is needed at this point to correctly assess the services rendered by the computing system so that charges can be levied on the basis of usage (the more resource used, the greater the charge).

Objective

The objective of this paper is to explain how university computing center billing algorithms evolve, based on the experiences of the

author in maintaining the software for computer billing at Mississippi State University for several years, and more importantly based on contributions of many of the MSU Computer Science/Computing Center staff members.

Goals of Computer Billing

Equitability in recovering a defined amount of money (budget) is the most important goal of any computer billing algorithm, once a cost structure has been established. Charges must be levied on the basis of usage. Equitability is easier to implement philisophically than it is technically. In the interests of equitability, it would be necessary to have sliding (rate) scales for the usage of core storage. I/O processing would be payable at a different rate than the CPU rate. Demand and real time processing would be done at different rates than batch processing. Different batch run priorities should have different rates. These are factors which introduce more technical complexity into the billing algorithm.

Another goal of a computer billing algorithm should be simplicity. The "bundle of services" provided by a large scale computing center is often sufficiently complicated without adding the various sliding scales imposed in the interests of equitability. People need to understand very well the various sources of computing charges. Sliding scales are much more difficult to explain to users, particularly those who are not technically trained.

An overall computing center goal which can be most effectively attacked by an efficient billing algorithm is the goal of making effective usage of the limited resources which are organized into a computing center. Much can be done to cause the center to provide as many services

as possible to the total user community, if the billing algorithm causes some real dollar charges to be charged to the users of all services rendered. The dollar is the only effective means of controlling the usage of limited resources in the opinion of this author.

Desirable Characteristics of a University Computing Billing Algorithm

a. The "Responsibility" Charge

Much of the computer usage in a university computing center is funded by the university administration. This usually includes educational and administrative usage, unsponsored faculty research and system overhead. Without careful management, the units generating the computer usage, usually educational or administrative departments within the university, can be totally divorced from the payment for computing services rendered. The "scarce resource" suddenly becomes a "free good." Most professional people have no reaction to this situation, but the small percentage that will take advantage of not being directly responsible to the administrative unit paying for the computing resources that they consume can cause too much of the computer resources to be devoted to very poorly planned "work", the usefulness of which is nebulous, and sometimes nonexistant.

It is therefore imperative that a token fee be charged such users so that they will be monetarily encouraged to use computer resources in a responsible way. The token fee should be payable in real dollars from departmental budgets. This concept is harder to "sell" when dealing with administrative and system overhead (computing center) units, but it is valid even in the case of system overhead. University users of computing facilities should be ever reminded that they are using scarce resources.

b. Defines enough user groups to allow general billing rates for each group. Also, there should be enough groups defined to generate computer usage reports which separate the various organizations using the computing facility (for example, private industry and government agencies might be billed the same way, but should be defined in different groups for effective bookkeeping) into homogenous groups.

c. Provides the capability to adjust (discount) all accounts as the need arises. Special situations create this need. For example, a professor can make a contract to provide university research to a government agency and grossly underestimate his computing needs. Although the professor should make a more reasonable estimate on the next contract, most administrators would probably allow him to complete his research, thereby generating the need for a discount.

d. Should clearly state the origins of all cost items, including the number of units of services used, the per unit cost, and the total cost. If sliding scales are involved, the "average" of the variables on which the scale "slides" should also be present.

e. Gives the user reports (if desired) in detail to the smallest cost transaction. This would be the individual run for direct computer usage. Other charge data such as programming or key punching should be dated and available to the smallest level of detail used to derive the charges.

f. Permits automatic collection of all computer performance data. The ease with which this task can be accomplished depends upon the raw system performance data provided by the operating system. All vendors of large scale machines provide billing information. It is

usually necessary that user programs be written to capture the "raw" information and build billing reports and invoices. Some system performance billing data is not available in the files provided by the operating systems. Examples include charges for forms changes, tape and disk mounts, and permenent mass storage usage (backed-up user files).

g. Allows adjustments to all cost units. There are many reasons for the need to make adjustments. Examples of this include: a system error is made in key punching billing data inputs; a printer loses print columns and causes a report to be rerun; the operator mounts the wrong form; pages were generated but not printed; and extra copies of the same print file were printed, to cite a few.

h. Provides for efficient processing of computer related services which are not the result of computer usage. Programming and key punching charges go in this category. Although these charges are trivial from a conceptual standpoint, they can involve heavy inputs of a bookkeeper's time. Forms should be designed that allow these charges to be recorded at the time they occur, and not recoded before inputting them into the billing system (punched into a computer readable form).

i. Should do automatic updates of all accounting files. This includes the posting of all payments, updating of balances brought forward and updating of year-to-date data files.

j. Provides the capability to collect historical data. Historical (sometimes hysterical!) data is needed for the development of annual reports, audits, and trend analysis needed to study future computing needs. These data should be generated in a form that will allow

each cost variable to be analyzed on at least a monthly basis. Great detail in historical data can become rather expensive.

k. Gives an estimate of direct computing costs at run time. Providing the user with run-time dollar costs usually requires EXEC charges or system generation parameters, or both. If the billing algorithm is not too complex, this information is a very useful service to provide for the user.

l. Has the capability to generate year-to-date or month-to-date billing reports on a request basis for individual accounts. Although computer billing systems are typically a serial, end-of-the-month batch processing job, the above capabilities are certainly to be desired. A careful analysis of the benefits enjoyed by inclusion of these capabilities versus the costs of implementing them should be made prior to designing them into a billing algorithm.

Billing Algorithm Development

The comments in this section are not intended to convey the thought that there is a way to specify a billing algorithm, but rather to give some ideas regarding how a billing algorithm for a large scale university computing center can evolve.

Initial inputs must come from the university administration. They tell the computing center director the percentage of his budget that they intend to fund. The computing center director then requests technical inputs from his people which are designed to predict the expected workload over the period for which the billing algorithm is to remain in effect, usually at least one fiscal year.

The workload should be defined in terms of the various user groups which could be paying different rates for computer services. It is very essential that estimates of this nature be as accurate as possible.

The director must then define his expected budget for the coming fiscal year. The projected budget should include all equipment payments, plus interest, maintenance costs, personnel costs, including not only operating systems personnel, but all personnel who are directing part or all of their work time toward providing computer services. Included would be systems, programming, and all other people who are needed to provide for a smooth running computing center which reacts not only to the intra-university user community, but also to as many outside users as necessary to fulfill the computing needs of outside people who are depending on university computing services. Outside business is an excellent way for a university computing center to defray some of the expenses of maintaining a large scale computing system, as long as it is remembered that the first priority of a university computing center is university service.

Next, the director can set the "responsibility charge." This is a token fee, which is intended simply to remind people that a computing system is to be used in a responsible way. This fee might be less than 5% of the rate charged outside users of the center. It should certainly be applicable to educational and unsponsored research users within the university. This concept is also useful if it is possible to apply it to administrative usage and system overhead.

Materials can best be covered on the basis of cost. Many computing centers make this a hard dollar item (users of computer services pay at least the cost of materials regardless of which user group they are placed in).

Mass storage (in the form of backed-up files) is a very dynamic area. The experience at the Mississippi State Computing Center has been

that the appetite of the user community for backed-up files has historically been very strong, and it is very easy for mass storage to become "saturated" several times, creating not only a need for more mass storage, but also automatic rollin/rollout of mass storage files to/from tape and adequate software to maintain backed-up user files on tape. Therefore, it is important that charges for usage of mass storage be set at least high enough to buy additional mass storage as the need arises. It is the opinion of this author that all mass storage charges should be payable in real dollars.

A concept that is theoretically sound, but very hard to implement from a practical standpoint, is charges for temporary mass storage usage.

Programming services, key punching and verifying services, and "extra attention" operator services such as forms changes, tape mounts, and the mounting of removable disk packs are also areas which require careful attention. Programming and key punching are relatively easy to justify. The more services used, the more people directly needed. These services can be charged to all users requiring them on a "cost plus" basis. Charges for special services performed by the operators are harder to "sell", but they also add to the number of people needed to run the operation. In this author's opinion, these services should be paid in hard dollars, with one exception. Other university people (outside the computing area) frequently need advice regarding how to solve their computing problems; if, say less than 30 minutes of a programmer's time is required to get these people started on the solution of their own problems, then it is this author's opinion that the computing center should absorb this cost. If consultation with other university organizations become very lengthy, and reach the "you do it all" plateau, then hard dollars should be paid for these services.

Up to this point, most charge areas concerning usage of the computing center by university personnel for university activities have been examined. As stated earlier, it is this author's opinion that some hard dollars should be budgeted for all university computer users including the university administrative usage and computing center overhead usage (systems, programming, etc.). Attempts should be made to stay within the budget defined, thereby encouraging all users to treat the computing facility as a limited resource.

Once tenative rates for the above charge categories have been established, then they can be applied to past performance data for all university departments using the computing facility. The administration can then make real dollar allocations which are earmarked for computing to all departments which are to have part or all of their computing costs underwritten by the university. These allocations could be perhaps 5% of expected CPU charges, but should cover all expected charges for materials, direct people services such as programming or key punching, and mass storage.

University departments performing research for government agencies or private industry through official university channels are generally charged somewhat lower rates than outside users or university personnel engaged in private consulting activities. One reason for this is that in engaging in funded research activities, university professionals gain in knowledge and expertise, which have spinoffs into their university duties and responsibilities.

Rates charged outside personnel should be set high enough that they do not compete with private organizations who provide the same services. A very happy situation for all is one in which university computing services are used by outside individuals or organizations who cannot otherwise obtain them.

From the standpoint of the computing center administration, dealing with outside organizations, including the federal government should always show a profit. University support of a "select few" outside the university simply cannot be justified by institutions of higher learning, especially those supported by the state.

Once rates have been set for all services, "fine tuning" is the next order of business. Expected revenues should be generated, to insure that the rates have been set high enough to meet the budget. When the expected revenues will meet the budget, more equitability can be built into the algorithm by examining sliding scales for some services and differential prices for others.

Sliding scales are particularly applicable to core storage usage. One way to do this is to set some lower limit (maybe 75% of the first estimate of the rate) on CPU cycles and let the core storage used "slide" the scale up to some upper limit of the first estimate of the rate (say 125% of the estimate). Sliding scales are certainly more equitable, but they are harder to explain to the user community.

Differential pricing comes into play when one compares I/O control unit cycles to CPU cycles. Although I/O operations require some CPU cycles, the cycles used by I/O control units should theoretically cost less than CPU cycles. Equitability demands that CPU cycles cost more, since I/O bound users would be paying relatively more for computing and CPU bound users relatively less under a system specifying the same rate for direct CPU cycles as that of I/O cycles. Differentials in this area are also harder to explain to the user community.

Demand processing is an area in which differential pricing cannot be ignored. Let's face it -- demand usage is priority usage. If batch rates

are based on priorities, demand rates should be higher than the rates for the highest batch priorities. Heavy demand processing makes heavy demands on computing facilities. Demand makes individuals more productive, but computing systems less productive, per unit of work. In the opinion of this author, monetary constraints are the only way to limit demand usage to a level that allows "reasonable" equipment budgets. Another useful concept in this area is the implementation of limited demand usage (low core load, reentrant processor) terminals, which are accompanied by lower prices.

Differential pricing for different batch priority levels is also an equitable idea. Priorities are a problem area for all computing center administrators. Technical problems associated with differential pricing by priorities are much easier to solve than political problems associated with priority determination. Technically, this can be defined in much the same way that the sliding scale for core utilization was described.

Summary and Conclusion

The establishment of computer billing algorithms for university computing facilities is a complex problem which should be attacked by careful planning. Groups having similar characteristics with respect to computer billing should be established, projected computing needs determined and hard dollar allocations made to cover some predefined portion of these users' expected computer costs. This would encourage intra-university users of computing services to utilize them as the limited resources that they are.

Rates for outside users would then be established and "fine tuning" done in the interests of equitability.

One conclusion is nearly as certain as death and taxes --- the algorithm established will not suffice. University people will run out of

computing dollars, inequities will creep in and there will usually be some unhappy people. These "bad omens" should not deter computing center management from carefully planned computer billing algorithms to the extent that the dollar is used to encourage efficient utilization of the computer resource.

A second conclusion is obvious to this author: without monetary constraints, the directors of large university computing centers can be certain that they will never have adequate computing capabilities.