FROM COST ACCOUNTING TO SERVICE LEVELS AND CAPACITY PLANNING: A PLANNED INTEGRATED APPROACH.

S. ORSINI - Bank of Italy

ABSTRACT

In 1992 the Bank of Italy needed a procedure to charge back the utilization of EDP resources of the mainframes (running MVS) to the end users. The base of this activity is the measurement of resource utilization (CPU, DASD, TAPES, PRINTERS) and the summary for specific different user groups for long term periods.

The measured resources were:

- CPU in Service Units;
- DASD in GigaBytes;
- CONTROL UNITS in Start Subchannel;
- Automatic Tape Library in number of mounts;
- Tapes in number of tapes;
- Printers in number of lines printed.

The chargeback procedure was developed to allocate the computer resource utilization to different user groups who are part of the Functional Units of the Bank of Italy.

Before starting with the project we considered the Capacity Management needs which also involved Capacity Planning, Service Levels and Performance Measurements. These activities were already running with different products and disomogeneous databases.

The base project of Cost Accounting was based on an architecture which was implemented with a flexible and modular approach. In 1993 the project was completed. In 1994 we implemented with the same architecture all the Capacity Management procedures. Now we have an integrated performance database based on SAS/CPE for all our needs, and we are thinking to implement the same concepts on the OPEN/VMS and AIX platforms.
This paper deals about the process of Cost Accounting and the subsequent implementations.

1. PREMISE

In order to manage computer systems efficiently it is necessary to measure the performance systematically. In the last years the measurement activity has gradually grown. Initially it was enough to monitor the systems performance; afterwards the concepts of service levels and capacity planning were introduced and the problem of information systems dimensioning and more recently the need to allocate the EDP costs among the final users (cost accounting) arose.

All these activities are part of Capacity Management (CM) which is based on the relevant systems performance measurement data that is aggregated and used in different ways for specific tasks. For example, in cost accounting we are generally interested in data aggregated on long time intervals (month or year), while for performance management we are interested in data of extreme detail which contains the single transaction occurrence or the single job.

The needs for performance management, capacity planning, cost accounting arose at different times and therefore were satisfied with specific solutions that are based on different software tools. Moreover, the system measurement is an activity always changing because its link to technology innovations and new application architectures.

In Bank of Italy, the opportunity for a systematic approach to this subject was the idea to allocate EDP costs among the users (Cost Accounting). In this circumstance a software tool was chosen (SAS/MXG) that was able to handle all the activities of Capacity Management through a flexible and modular structure. This product allowed us to implement gradually the Capacity Management project and satisfy new demands, which initially weren't expected, integrating them in a coherent way in the existent architecture.

The present work describes the main elements of the Cost Accounting project and its following implementations that have been accomplished for the certification of the service levels, capacity planning and performance management.
2. THE DIVISION OF THE EDP COSTS

In the last years the investments in the EDP area have remarkably grown. At the same time the users of the information services have become more demanding in terms of service quality (availability, reliability, etc) which made it necessary to support always greater costs in order to maintain the users high degree of satisfaction. The users awareness of the computer processes cost is an important feedback that enriches information systems development. The accounting process of the EDP costs and the following allocation among the users of the computer services is called COST ACCOUNTING. The base of this activity is measuring the consumption of computer resources (CPU, Storage, I/O) in a structured way (according to predefined groups) and aggregating them for long time intervals. The consumption collected produces synthetic information for an accounting procedure which uses a table of costs (owned by the accounting procedure) and consumption data (acquired automatically by the system measurements). In the following, the fundamental elements that are exclusively of the system measurement will be described.

3. OBJECTIVES OF COST ACCOUNTING

The collection activity of computer resource utilization needs a product able to store performance data and perform analysis, and that has integrated functions that allow to satisfy specific needs in a simple way.

The main characteristics a measurement system must provide are:

- high flexibility;
- the ability to acquire automatically the measurements from all the collecting tools;
- the capacity to acquire information manually (operative way);
- graphic features and statistics;

The preliminary activities for the implementation of an accounting system are:

- determine the resources and the criteria of evaluations on their use depending on the available measures;
- determine the consumers and their functional groupings to define the division's criteria of the consumption.
3.1 RESOURCES

The resources must be easily measurable. When the number of resources grows the accounting system becomes more complex. Therefore the choice must be coherent with the company's needs and objectives. In our case the computer resources that respond to these needs are:

- the central processor (CPU);
- the magnetic disk (DASD) and the relative control unit;
- the magnetic tape and the relative control unit and automatic tape libraries;
- the printers.

The consumption collection is done on a daily basis and then aggregated in a longer period (one month). The collection must indicate the amount of resources used by each consumer. The measure units must have the requisite of being invariant with respect to the surrounding environment (hardware and software). The most adequate measures and the corresponding to measurable units in the MVS environment are:

- for the central processor unit: CPU in SERVICE UNITS\(^1\);
- for the magnetic disks: the space occupied in MBYTES;
- for the disks control unit: the I/O activity in SERVICE UNITS;
- for tapes: the number of tapes;
- for the tapes control unit: the I/O activity in SERVICE UNITS;
- for the automatic tape libraries: the number of mounts;
- for the printers: the number of lines printed (if the prevalent printing activity is on-line).

The measurement can be done in two ways:

1) using data collectors always active;

2) taking a specific measure.

\(^1\)Service Units (SU): a measurement unit used by the MVS operating system to account the use of resources (CPU, storage and I/O subsystem). This measurement unit is invariant in respect to the computer power of different systems.
For example, the second type is used to evaluate allocation of magnetic disks and tapes. This is a global measurement taking which is done periodically (for example once a month), neglecting the movements between two measures. For all the other measurements there are monitors always active in the system.

3.2 THE CONSUMERS

The consumption collection must provide the allocation of resources used directly by each consumer and divide those which are indirectly used. Moreover, for a better control it is useful to determine for each consumer the tributed consumption among the different applications. The possibility to determine this level of detail depends on the organization standards.

The resources needed to manage the system (operating system, generalized software, management monitor, auxiliary products, capture ratio, etc) must be also considered in the consumption calculations.

Additionally, there is a percentage of resources which is not used for operative management needs.

We chose to make a proportional allocation of the system resources by each user which is the indirect consumer of these resources (the objective is to supply a high quality service).

4. MEASUREMENTS AND STANDARDS

The distributions of direct consumption can occur with major or minor difficulties depending on the standards adopted by the company (naming convention of transactions, terminals, users, etc). The insufficiency in standards generally involves an increase in work, and in certain cases could prevent from performing such a work. The investment in standards allows an efficient use of the measuring products and gives also an advantage for the future.

Standards should allow the types of distributions examined before, but also must guarantee simplicity in the aggregation. For the objectives of cost accounting the prerequisite on any standard is that both the applications and the functional groupings of users among which the costs are allocated are decoded. Following on, we will indicate the code of the application with APPL and the functional grouping with RAGG.
4.1 BATCH ENVIRONMENT

The batch work can be classified in two typologies:
1) Production work that is activated periodically
2) Extemporaneous work submitted directly from the TSO users.

For the first typology it's necessary to include in the working accounting fields the application code (APPL) and the user. The coding of the user is needed to decode the functional group (RAGG).

The second typology is subject to a larger flexibility and therefore is more difficult to control. It's possible to make the same reference as the TSO users name (opportunity coded) and the TSO accounting code, provided that the information at the moment of the job submission is dynamically acquired from the analogous information of the TSO user.

4.2 INTERACTIVE ENVIRONMENT (CICS, IMS)

The insertion of the application code (APPL) in the transaction identification and the coding of the functional grouping of the consumer (RAGG) using the terminal allows the maximum degree of analysis and synthesis.

4.3 INTERACTIVE ENVIRONMENT (TSO)

The TSO user identification and the use of the TSO accounting, occasionally different for each application used, provide the necessary details. Remember that the accounting code is specified when opening a TSO session and remains valid for the duration of that session. The user must set the corresponding accounting code of the application with which he intends to work (APPL) and if he wants to work with a different application he must close and open the session in order to change the accounting code.

This formality is not very pleasant for the consumer. An alternative could be the implementation of a function that allows the modification of the accounting code dynamically without closing the session. This function could be activated automatically for structured applications or directly by a user command.

4.4 DATA BASE MANAGEMENT SYSTEMS (DB2)

The resource consumption system in the MVS environment is such that the CPU consumption of DB2 is automatically attributed to those who request its services.
Depending on the connection (batch, IMS, or TSO) the debit is referenced as described before. To distribute the I/O activity of DB2 it's necessary to determine a representative indicator of the I/O activity (for example the number of get pages) to use and distribute the total I/O service units consumed by DB2. The association with the functional group (RAGG) and the application (APPL) can be identified for batch work and TSO activity because all the key values are provided within DB2 accounting codes. For the IMS activity, the "working plans" which perform the access to DB2 are named with application codes that allow to determine the application (APPL) while the user's name allows to determine the functional group (RAGG).

4.5 STORAGE DEVICES (DISKS, TAPES)

The evaluation of the space occupied on disk or tape can be performed by examining the data-sets. For the disks, the VTOC indicates the amount of space allocated for each data-set. For tapes, the data of the automatic tape library can be used to calculate the number of cartridges allocated. In both cases it's necessary that the names of the datasets archived on disk or tape contain the project code to which they refer (APPL).

For space shared by applications or users, for example system data, the principle of allocating proportionally the consumption is valid as previously described.

5. RESULTS IN SYNTHESIS

The analysis performed highlights that the complexity of an accounting system depends on the degree of effectiveness of the standards adopted in the nomenclature of the computer information entities.

In general it's not always possible to adopt all the hints exposed in the preceding paragraphs due to specific demands or to an existing historical structure whose modification would require an onerous work.

When standards are insufficient or don't exist at all, there's no possibility to identify APPL or RAGG. It is necessary to use 'exceptions tables' in which the main inconvenience is the burden of maintenance.

An accounting system must be based on reasonable approximations and processes of extrapolation that make it as easy as possible. In fact it was verified that the raising of precision, also even at a minimum level,
sometimes requires a remarkable increase in the system complexity which is definitely not convenient.

6. THE FOLLOWING DEVELOPMENTS

At the end of the accounting project the activity was carried on with the objective to replace the software tool that performed collection, management and analysis of the used system performance data and service level certification procedure. The service level procedure that was released in 1987 had notable limitations in the used software tools and in the application logic that were evident in time. The first goal was the migration to the same tool used for cost accounting (SAS/MXG) and, where it was possible, using the same data.

In fact the base of data previously used for capacity management were built in different times due to the different operating needs. For this reason there were disomogeneous characteristics and different tools used. Data redundancy was notable and complex to manage.

The migration project to the new software tool allowed to perform an analysis that has interest beyond the service levels. The problem of rationalization and integration of the necessary data bases for all the capacity management activities was analyzed.

The use of the new integrated data base allows:

- the utilization of the same tool and the same methodology for all measures;
- the elimination of data redundancy;
- the simplification of data base management;
- the coherence of performance and cost accounting data.

The predefined structure and the portability of the software tool (SAS) make it possible also to integrate data from different platforms in respect to the traditional mainframe.

7. SERVICE LEVELS

The certification of service levels is built on the basis of a preventive contract with the consumer that receives the supplied services. The quality of the service
level is evaluated on behalf of the timeliness\(^2\), reliability\(^3\) and the performance of different computer applications.

The evolution of the technology in the years has left partially obsolete some techniques of measurement of the original procedure. It was necessary to add additional criteria of evaluation of the quality of the services provided in order to let them adhere to the new reality.

The service level agreements must be compatible with the typology of the work.

The choice of the type of measure depends on the type of execution for the work. The average response time is extremely important for transaction and interactive activity; vice versa, for batch jobs the timeliness and reliability are of major importance.

Also the characteristics of the logical environment in which the work is executed engage a fundamental importance on the service levels (different needs for an experimental environment compared to those of development or production environment).

The definition of service levels must be coherent with the application architecture and technology, therefore it's important to avoid:

- to contract an impossible service level;
- to certify a service perfectly respondent to the expectations of the user with the statement "not satisfactory enough";
- to certify a service not respondent to the expectations of the user with the statement "satisfactory".

8. THE CLASSIFICATION CRITERIA

The previous classification of the IMS transactions were based on the number of accesses to the hierarchical DL1 database and didn't take into account the requests to the DB2 relational database. Therefore there was a bad estimation of the transaction weight. For example, a transaction that required a lot of DB2 services without accessing DL1 was classified as "LIGHT" even though it consumed much more resources.

\(^2\) Timeliness : the relationship between the number of works started and concluded in the expected time and the total number of works executed.

\(^3\) Reliability : the relationship between the number of works terminated correctly and the number of total works.
In such circumstances achieving a performance objective (i.e. response time lower than 3 seconds) could result impossible.

On the contrary a transaction that made lot of calls to DL1 and therefore classified as "HEAVY", thanks to the evolution of the I/O architecture that provided always elevated performance could achieve the performance objectives of a light transaction.

The larger spreading of the relational technique gradually made the initially adopted classification criteria inadequate.

The new criteria is based on CPU utilization (measured in Service units), using the same logic of the TSO environment.

Such criteria allows to take into account, in a simple and correct mode, different database management systems (hierarchical and relational).

Another modification introduced in the transaction environment was for complex transactions for which it was expected a performance objective < 30 seconds and no matter what the absorbment level was.

To accept an indefinite weight on complex transactions and at the same time set a performance objective < 30 was risky, in particular in the relational world where response time grows depending on the amount of data accessed.

Another aggregated class was defined for complex transactions that didn't have any performance objectives. In this class are the transactions that absorb a number of service units that pass pre-established thresholds (higher than 500 SU). The old classes were:

- light transactions: CPU < 40 SU, response time < 3 sec.;
- medium transactions: 40 SU < CPU < 100 SU, response time < 6 sec;
- heavy transactions: 100 SU < CPU < 500 SU, response time < 30 sec.

9. INTEGRATED DATABASE

Enclosed with this document we have the design of the integrated database that contains all the data of interest for capacity management. The first page reports the service level data, the second reports the relative performance control data and the third contains the relevant
information for capacity planning. The data is fully integrated and is generated by a common area that contains the information on the IMS transactions (IMSTRAN) and the users regions (TSO, batch). Subsequent information can be added with extreme simplicity.

10. CONCLUSION

This whole project (2 years of duration) has achieved the following objectives:

- the implementation of the Cost Accounting project;
- the replacement of the software tool for service levels procedure;
- the rationalization and integration of the data base necessary for all the capacity management activity (capacity planning, performance management and cost accounting) in the mainframe environment.

The prearranged structure and the portability of the software tool used make it possible to integrate information from different platforms, for example Digital and RISC/6000, with the traditional mainframe. In fact, we are now implementing a project which will allow to collect performance data from the Vax, Alpha and RISC/6000 platforms using SAS and SAS/CPE.