IBM

Washington Systems Center

Technical Bulletin

Capacity Planning Overview

R. M. Armstrong

GG66-0254-00
July 1986

Washington Systems Center
Gaithersburg, Maryland
Technical Bulletin

CAPACITY PLANNING OVERVIEW

R. M. Armstrong

GG66-0254-00
July 1986

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis **without any warranty either expressed or implied.** The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

It is possible that this material may contain reference to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country. Such references or information must not be construed to mean that IBM intends to announce such IBM products, programming, or services in your country.

Publications are not stocked at the address given below; requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to: IBM Corporation, Washington Systems Center, 18100 Frederick Pike, Gaithersburg, MD 20879

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation whatever. You may, of course, continue to use the information you supply.

Abstract

A key to a successful installation is capacity planning. The approach to capacity planning is significant -- especially with multiple systems and on-line applications. Many installations can no longer navigate by their wake. Topics discussed include business requirements, DP implementation and service time, and capacity planning system design.

A more detailed treatment of some of these techniques can be found in Capacity Planning and Performance Management Methodology, GG22-9288-0.

Acknowledgments

The capacity planning topics and techniques in this bulletin are a result of working with many IBMers and IBM customers on their needs in capacity planning and performance analysis.

Table of Contents

INTRODUCTION

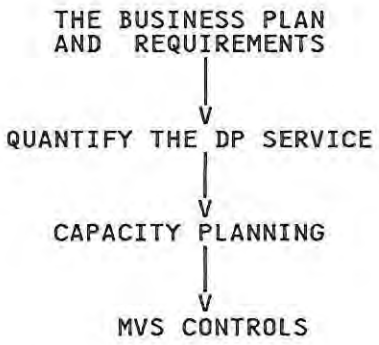
Today's data processing environment for a business enterprise includes large systems, on-line systems, multiple systems, and multiple sites. In more and more cases, the competitiveness and even survival of the business depends on quick access to timely information. Capacity planning can help grow the DP installation and evaluate alternative choices in support of the business needs. In the past, capacity planning has evaluated trend lines of past performance. For many, "navigating by your wake" no longer works. The DP workloads are being created and expanded by non-DP end-users in large terminal networks. More employees are going on-line. In some cases, customers and vendors are going on-line. All this means that more powerful data processing systems are required. Programs to service these "non-DP end-users" are becoming more "user friendly". Again, more data processing power is required.

Data processing is an integrated part of how the company does business. The dominant question today is not how little it can cost -- although that is still a factor -- but rather what does it buy me. People productivity on the right stuff is the most important guideline for capacity planning. The "right stuff" is usually determined from business planning data. The capacity planner generally can not evaluate that the user is doing the right things, but the capacity planner can evaluate the system performance aspects of user productivity. The main purpose of this paper is to review and discuss the scope of capacity planning.

Capacity planning includes knowledge of user requirements, performance analysis, and system design. A general knowledge of what is important to the user can be useful toward providing the service that is needed. Then, more specifically, response time and transaction rate make up user requirements. Response time is a component of user productivity; transaction rate is a measure of number of users and business volume. By organizing the performance data in terms of logical groups of users, capacity planning becomes a system design process. On a broader scale, the capacity planner should keep track of the user's end-products. This is more difficult, but it is the "bottom line" evaluation. For example: Are the application development projects on schedule? How many "widgets" are being sold because of DP support? How low are the costs of manufacturing and distribution because of a product information and control data base? How quick is the time to design a new product because of DP support? How much better is the quality control system with DP support? Understanding these factors gives management a picture of the DP support business. Otherwise, DP is just another cost center.

Capacity planning topic areas include the business plan, quantifying the average service for transactions in a workload (logical group of users), capacity planning techniques, and MVS controls. The business plan and business requirements are a very important part of capacity planning. Business requirements are the basis for knowing that we (the DP installation) are supporting the right stuff. The business plan implicitly specifies logical groups of users and provides a structure for measurement and analysis. This structure, in DP terms, is often referred to as a business element structure.

TOPICS:

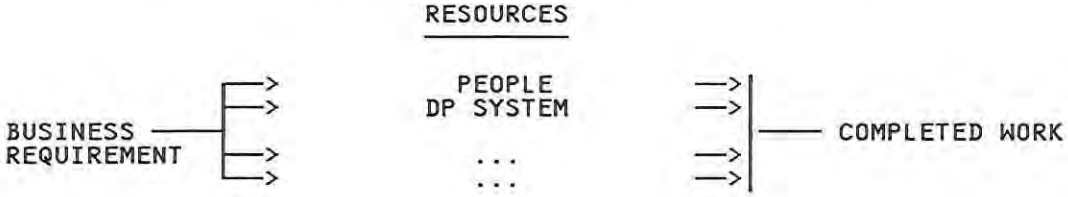


Quantifying the DP service is a technical task of using measurement data and data reduction techniques to evaluate the resource required by each workload. In the past this often meant evaluating the utilization for a given transaction rate. Today with more sophisticated analysis techniques this means evaluating the average amount of resource per transaction required by each workload. Then the average service together with the required transaction rate can be used to analyze utilization and response time for alternative combinations of workloads on several possible system(s) configurations.

Capacity planning also includes topics on contingencies, availability, and recovery. The capacity plan should include controls on resource consumption so that the planned objectives are met. These controls must be specified so that the system is managed in a consistent way.

WHAT IS CAPACITY PLANNING?

Capacity planning is (1) relating business plans to data processing workloads and performance requirements and (2) estimating the DP system that can support these workloads at the required performance. In the total picture, capacity planning must also include costs and revenue. There is nothing unique to dealing with costs and revenue because of capacity planning techniques. (Costs and revenue are referred to in this discussion but are not treated explicitly.)



CAPACITY PLANNING

Why do it?
Why is it important?
What is the capacity planning effort worth?

This is not a discussion of whether or not to do capacity planning. Actually, one way or another, everyone does capacity planning. If you have ever upgraded a system component, you have done capacity planning. Some do it in a more timely fashion than others. Some relate it more explicitly to business opportunity than others. Some do it when the controller says ... we have got some money now.

As more employees go on-line and the DP system becomes an integral part of their job, the importance of capacity planning increases. Now we are talking about the productivity of that employee multiplied by the number of employees doing that type of work. Now we are talking about the volume of business. Another factor that makes the capacity planning discipline important is the growth and change that is happening at many installations. In a dynamic environment "formal"

capacity planning can help identify alternatives that are possible and avoid alternatives that do not work.

Do you know what your capacity planning effort is worth? "Too much capacity" means more DP cost dollars. Cost can be a complex topic. Being over-capacity because you just bought some equipment in the early part of its (technology) life cycle may actually cost the business less. "Too little capacity" is fewer DP cost dollars but can translate into less revenue on the one hand and more cost through less productivity on the other. There probably is more leverage and exposure for the under-capacity case. What is a dollar invested in DP worth? What are your exposures?

How is the DP organization viewed as a business center? How does this relate to supporting the business plan? Many DP organizations are viewed as cost centers. Back when the user was isolated from the DP system, the DP budget was small but growing almost arbitrarily at 30%, the cost of hardware components was decreasing, and a cost center view worked. With the cost center the user tends to be not accountable and may just do less since it is not in the budget. The installation tends toward minimum service as the business grows; after all, they provided everything the budget allowed. The cost center approach is not consistent with supporting the business plan. It is too loosely coupled with the business needs. Fortunately, many of these "cost centers" act, to a large degree, like profit centers. With the profit center, there is potential for accountability on both sides if the value of a transaction is determined. This is not an easy task and will not be solved to your satisfaction on the first try. The profit center relates most directly to production on-line workloads. Workloads like program development are more difficult and an info center workload may be the most difficult. It may be appropriate to treat the info center like "research" at this time. Allocate a certain amount of resource and live with that amount. (Some companies allocate a particular percent of their revenue to their research.)

A few DP installations are viewed as service centers where the goal is to never run out of "cycles" that a user may want. When you consider the leverage of the DP support, this may be a good way to do it. It simplifies the arguments and certainly has a productivity focus. In the service center case DP is not accountable.

Back to the profit center. The profit center implies that DP knows the business economics in terms of end-products -- to some extent. Isn't that a good idea anyway? DP should also understand the relationships between user productivity and system capacity. If capacity has no effect on productivity, you do not need the system(!). If capacity has a small effect, consider a cost center approach. If capacity has a dramatic effect, consider a service center approach. If it is a real trade-off, consider a profit center approach.

TECHNIQUES & SKILLS

What is your capacity planning philosophy? Problems or opportunities? Some have a problem-type philosophy. Start to evaluate with symptoms of "things" not working. (If it ain't broke, don't fix it.) Then when something is not working, fix something. Next measure and see if you fixed the thing that was broken.

Consider an opportunity approach. Start with a "theory" or "flow" of transactions in the system. Design the system to support the business requirements. Measure the results and manage the system. Design based on business requirements is the underlying philosophy in this paper.

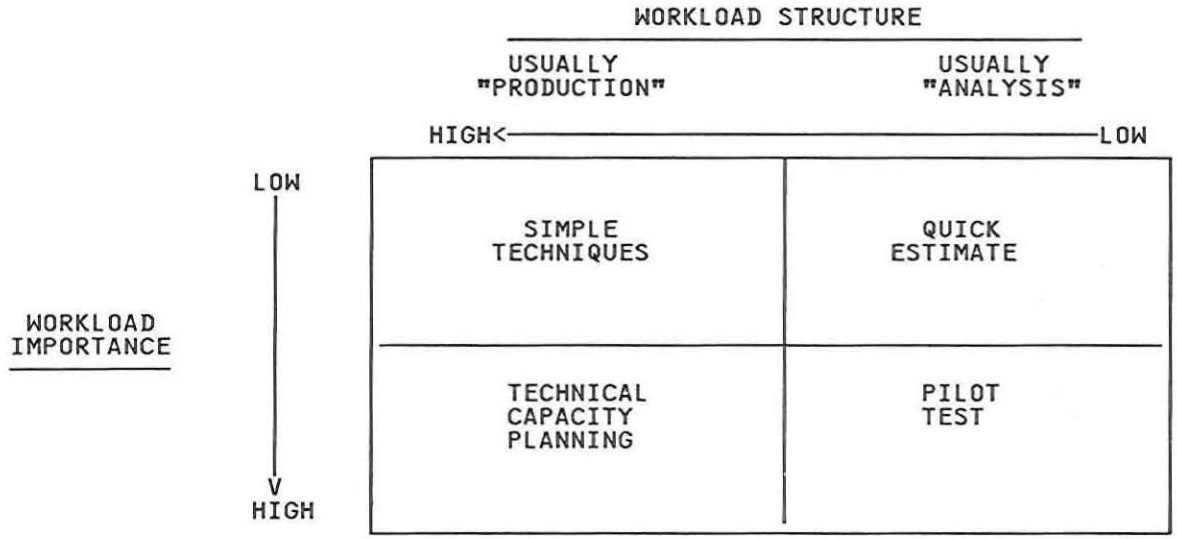
There are a number of techniques used in capacity planning today that are different from earlier alternatives. In the days when the operating system was on tape (!) some nice linear programming type techniques were starting to come out when all of a sudden the operating system was on disk and everything was different. Next, thruput techniques were developed and the focus was on CPU utilization. Unfortunately the users and applications did not stand still and more and more users went "on-line". As workloads grew, users started complaining even though their work was getting done ... at 90% CPU utilization. System programmers began mumbling about "DASD contention", "paging", and enough "real storage". Out-of-town specialists started talking about "rule-of-thumb" utilizations. Installations found that different users had different requirements and that response time could be important. But response time and utilization were non-linearly related. Phrases like "we must have hit the knee of the curve" are now heard at the coffee break. "Response time" and "service level agreement" are becoming household words.

The introduction of business elements is probably the most significant "new" technique. This began in the mid to late 70's. Resource consumption by itself without a breakdown by business element is not adequate for many capacity planning situations -- because different parts of the business are changing in different ways. Planned growth in many of today's system needs to be driven by business element projections rather extrapolating past consumption.

Another technique that is required for analyzing response time is the breakout of transaction rate and service time as individual variables. Their product is utilization. Even if you do not analyze response time itself, knowing whether a change in utilization was due to a change in service time or a change in transaction rate is helpful. Alternatives for "fixing" service time are very different from alternatives for "fixing" transaction rate. (You do not "fix" utilization.)

Today's data processing systems are actually queueing systems. This means that in many cases statistics and queueing models can be used to help analyze and project performance (capacity). In the old batch systems, performance was determined by the detailed characteristics of one or a few jobs and queueing models were of little or no help. Some "overnight batch" requirements still fit this mold.

Capacity planning has several faces. The level of analysis should be suited to structure and importance of the workload. It does not make sense to do an extensive analysis on an unimportant workload that has questionable data. For workloads that "run the business" whatever analysis is required to get the answers is the level of analysis that makes sense.



EACH QUADRANT REQUIRES TOOLS AND TECHNIQUES FOR DATA COLLECTION, DATA REDUCTION, AND DATA ANALYSIS.

For an important production workload (often IMS or CICS), "technical capacity planning" may be appropriate. Business plans and response time to the end-user should be a part of the analysis. For a less important workload, a utilization and trend analysis may be sufficient. For a new important workload with fuzzy data and fuzzy user requirements, a pilot test may be the right answer. Common sense plays a significant role in capacity planning. You are allowed to use common sense in deciding what to analyze and to what depth.

Data collection tools and techniques are usually an integral part of the particular operating system -- such as the SMF records in an MVS system. For data reduction and data analysis, there are lots of individual tools. Choose tools and techniques that are commensurate with the capacity planning job to be done. You may want to write some code yourself to tie some of the output together. Net reports on the data that is important to your operation may be very helpful (i.e., productive). Other considerations include a performance data base and business element tracking. Tracking business element drivers will require an extra effort. They are probably not in an SMF record. Benchmarks

are seldom used today because it is not practical to benchmark the total system, and the purpose of a benchmark is (was) to learn how things "really" fit together in the total system.

SPECTRUM OF
PERFORMANCE ANALYSIS TECHNIQUES

A RESOURCES	GUIDELINES LINEAR ANALYSIS	USAGE HAND ANALYSIS CPX SLR AEC COURSE
TIME	QUEUEING	ISMI COURSE RESQ
COST	DISCRETE SIMULATION	
V	BENCHMARK	ACTUAL SYSTEM TPNS SYNTHETIC JOBS

NOTE: See your IBM Representative for additional tools.

Who can do capacity planning? What are the skills? Many of the tasks are the same old tasks that installations have been doing. These include collecting and analyzing measurement data for hardware, software and systems. There are some newer tasks. Working with business planning and user requirements is an important part of capacity planning. A capacity planner should develop a general knowledge of what the users are doing and what facilities in the DP system are important to them. A little common sense sometimes goes a long way. The capacity planner also needs to work with the user groups to develop service level agreements -- we will come back to this later. Still another area is performance evaluation for application development. More than once, application development performance estimates have fallen short of what was needed and what was practical, jeopardizing the whole project. Queueing is a relatively new discipline to DP support organizations. Consideration should be given to upgrading skills in this area. System measurements, business planning, and modeling are basically three disciplines. The best approach may be three people. Individual differences can be so great that each installation needs to decide what scope of capacity planning is right and what individuals are right.

There are several basic concepts in capacity planning. First, there really is a business plan. Some installations are so isolated that they have no detectable relation to the business plan. This should be changed. If the users know when the system is down, then we (the installation DP support) should know what that means to them in their terms. Indeed, there is a business plan, and, if we are not part of the plan, we are probably part of the problem. Average transactions do not have response time requirements. Business elements have response time requirements -- otherwise we would not have to complete the transaction. Business elements direct how the data should be structured. Data by business element allows us to talk about expected values for transaction rate, service time, and response time. Expanding these topics, however, goes beyond the scope of this paper.

Basic queueing systems topics include:

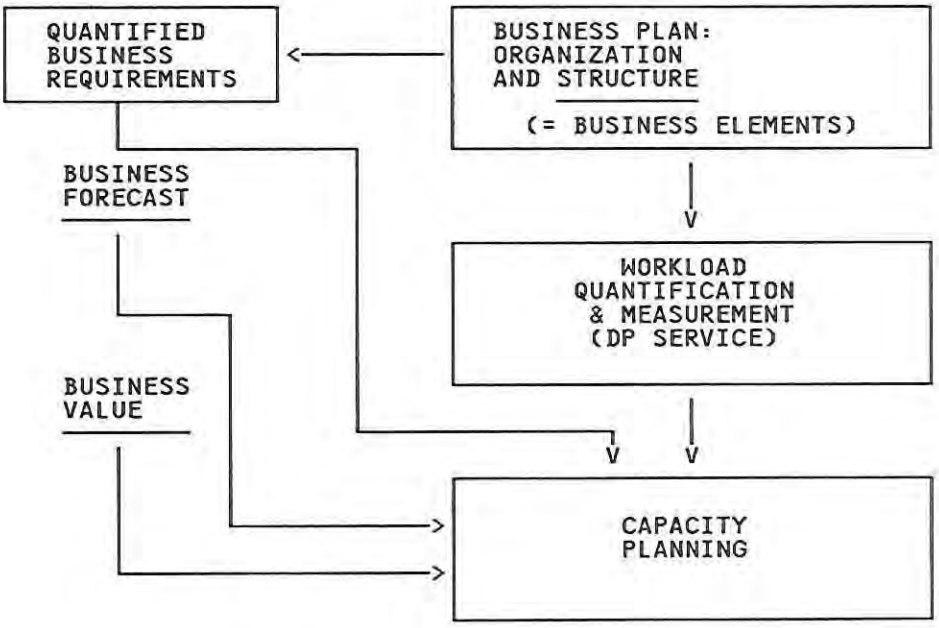
Probability density function
Probability distribution function
Moments of a distribution
Mean, coefficient of variation, variance, standard deviation
Little's Law
The concept of independent (vs dependent) events
Exponential distribution, Poisson process and distribution
Erlang density functions
M/M/1, M/G/1, and M/M/M server equations
The concept of closed system (limited queue or population)
Jackson's theorem (network of single servers with exp. distr.)
Stationarity (of a distribution)
Random process
Expected value

Knowledge and skill in these areas can help you analyze your on-line DP system.

THE BUSINESS PLAN

The business plan is the place to start capacity planning. You may want to jump right in and deal with the CPU or DASD details. Don't. The business plan provides a necessary organization for collecting data and evaluating alternatives. We do not need to know the entire business plan. We need to know the items that relate to DP support for the plan. We need to know about growth and change for projects and employees that use DP resources. We need to find the parts of the business plan that relate to specific major projects (or departments) or to end-products produced by the business where DP support is required. Call each of these logical parts a **BUSINESS ELEMENT**.

The business element structure is used for quantifying the performance related business requirements and the corresponding workload measurements.



What is the business value?

From a point of view of the business plan, how is the output or value of each business element measured? There must be a way (however fuzzy) of measuring a business element. For end-products the measure is the number of end-products. This number may need interpretation but at least it is a number and you can write it down. Some quantification of the business value allows us to evaluate at least the incremental changes in DP support facilities. Did it help? Did it hurt? What happened?

It is important to know the effect on the business, not just on the DP internal variables. For example, there was a case where giving an insurance claims adjuster better response time caused the adjuster to review the claim faster with a net result of payments for less than qualified claims. What are the business objectives for the end-user?

The real forecast is the business forecast. We want to find a key variable (or a few variables) for each business element that relates as directly as possible to DP resource consumption. Call this a **BUSINESS ELEMENT DRIVER**. Most of the time the DP resource consumption will be proportional to the business element driver. However, inversely proportional relations are possible. The object is to find a variable with a cause-and-effect relation. High correlation (by itself) is good, but if there are not underlying causal relations, we may be heading for a big surprise! Business element drivers allow us to forecast DP resource consumption based on business plans.

There is no mathematical formula for identifying business elements. Sometimes they will appear as line items in the business plan. If a business element can be identified by relating DP support requirements to an end-product, that is often the preferred choice because business element drivers and business values are easier to find.

IDENTIFYING BUSINESS ELEMENTS:
> LINE ITEM IN BUSINESS PLAN.
> RELATE TO AN END-PRODUCT.
> RELATE TO A RESPONSE TIME REQUIREMENT
- MULTIPLE "WORKLOADS" WITHIN A BUSINESS ELEMENT.

Some business elements need to be divided into separate parts because we want to manage each part differently. For example, the accounting department may be a business element with requirements for "interactive analysis" and for "overnight batch". Obviously, we want to manage the interactive analysis to a short response time and the overnight batch to a long (overnight is OK) response time. Let's call these parts **WORKLOAD**s within the business element.

Identifying all the workloads in the business may be possible. Is this desirable? Probably not. You must decide. After 90% of the CPU resource is identified, consider "lumping" the rest together. An exception might be a small but rapidly growing workload that will be important in the foreseeable future. The CPU is the focal point here but other resources are not ignored. A resource other than the CPU may be the "critical" resource. However, since "everybody" shares the CPU, the CPU is usually the resource to key on. How much granularity is needed to make decisions on capacity for you? That amount or perhaps one step more is probably the right amount to collect and analyze.

BUSINESS PRODUCTIVITY:
> REVENUE PER INTERVAL.
> COST PER PRODUCT.
> END-PRODUCTS PER INTERVAL.

Measuring the output or value of a business element is a difficult task. Many DP support organizations do not track this data. It must be done somewhere in the business -- else how do you manage the business? Some of the business variables that the DP installation should affect are revenue per interval, cost per product, and end-products per interval. In some cases the business value is in quality or some other hard-to-measure variable. Consider including internal deliverables for some of these fuzzy variables. For example, if we manufacture widgets and want to measure the publications department, what do we measure? Cost per product for a given level of documentation standards is one variable. But what about those standards and what about the quality (of content) of the manuals? Well, no one said it would be easy. Speaking of quality and standards, perhaps DP can provide some "decision support" models.

In summary, what is supposed to happen because of each business element? (If nothing is supposed to happen, we do not need the business element.)

The forecast is the business plan forecast for the business element driver. Many installations avoid this approach and extrapolate past utilizations. This is called "navigating by your wake". It works as long as we have a "business as usual" situation. It is also easier to do (by a wide margin). On the other hand, if we want to manage a dynamic and changing DP environment in support of changing business needs, the concept and reality of business element drivers is significant.

What key "measurables" are in the business plan for this business element? These are the business element drivers. Look for an end-product. If the resource consumption relates to the number of end-products, that is the easiest case to analyze.

BUSINESS ELEMENT DRIVERS:
> END-PRODUCT.
> HEAD COUNT.
> OTHER.

DP VARIABLES:
> CORRELATE TRANSACTION COUNT.
> CORRELATE NUMBER OF USERS.
> CORRELATE SERVICE/TRANSACTION.

Where head-count is key, the situation is almost as straight forward. There can be wide differences between two individuals, but when there are a number of people in a group, the output of the group is often predictable.

Business element drivers usually correlate to transaction counts or transaction rates. Note that head-count can be translated into transaction rate. (More on this later). Sometimes the correlation is to the service per transaction. For example, changes in number of employees or in the company benefits may change the service time of the payroll program. But this program will not be run more (or less) frequently.

BUSINESS REQUIREMENTS:
> RESPONSE TIME
> TRANSACTION RATE

(for each WORKLOAD)
> PRIORITY
> AVAILABILITY

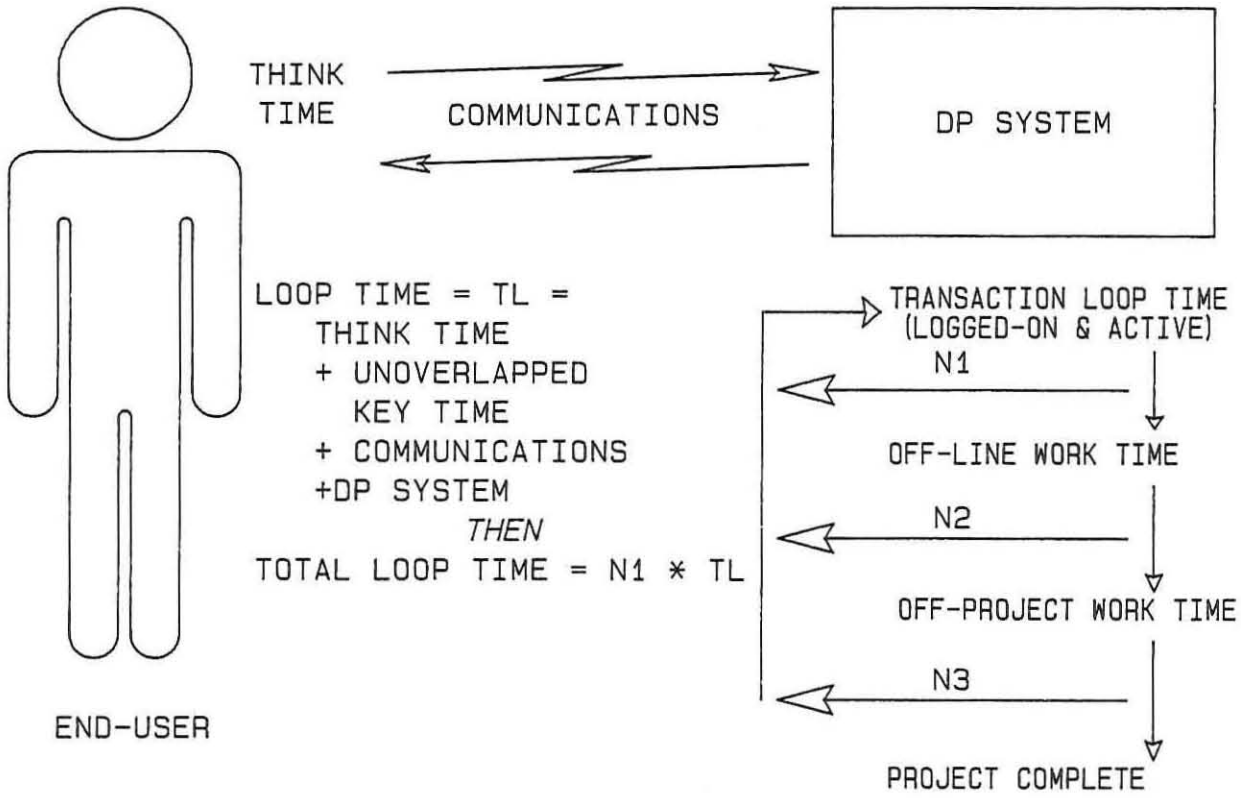
Capacity planning needs quantified business requirements. These include response time, transaction rate, availability, and priority. These requirements are to be evaluated over the planned period. As a minimum, the planned period includes the time to procure system components (such as a CPU). In some cases the planned period includes buying some real estate and putting up a building. The set of quantified requirements may be developed into a "Service Level Agreement". This is a document of understanding and commitment between the users and DP support regarding the performance that can be expected for given levels of user activity.

The term "transaction" is used throughout performance discussions. Let's pause to ask "what is a transaction?" Candidates include RMF transaction, function point, and user end-product. A user may appear more (or less) productive in terms of RMF transactions independently of user productivity measured in "real transactions". For changes (improvements, of course) over a short time period, measuring RMF transactions per user per interval should be a valid indicator of productivity. Fundamentally, productivity is the end-products or deliverables that the user produces during the time interval. Different deliverables may have different magnitudes in terms of work done or quality. Therefore equal rates (numerically) are not necessarily equal productivity to the business. In an attempt to avoid the potential pitfalls of RMF transactions and of variations in end-user deliverables, some installations have pursued "function points". A function point is a unit of work for a business element. At least that is this author's interpretation. Defining and tracking function points is not an easy approach. A practical approach at this time is to use RMF transactions as the metric and apply common sense to avoid occasional problems where shifts in the RMF transaction counts may be misleading. For example, going from batch to TSO, or going from line editing to a full screen editor.

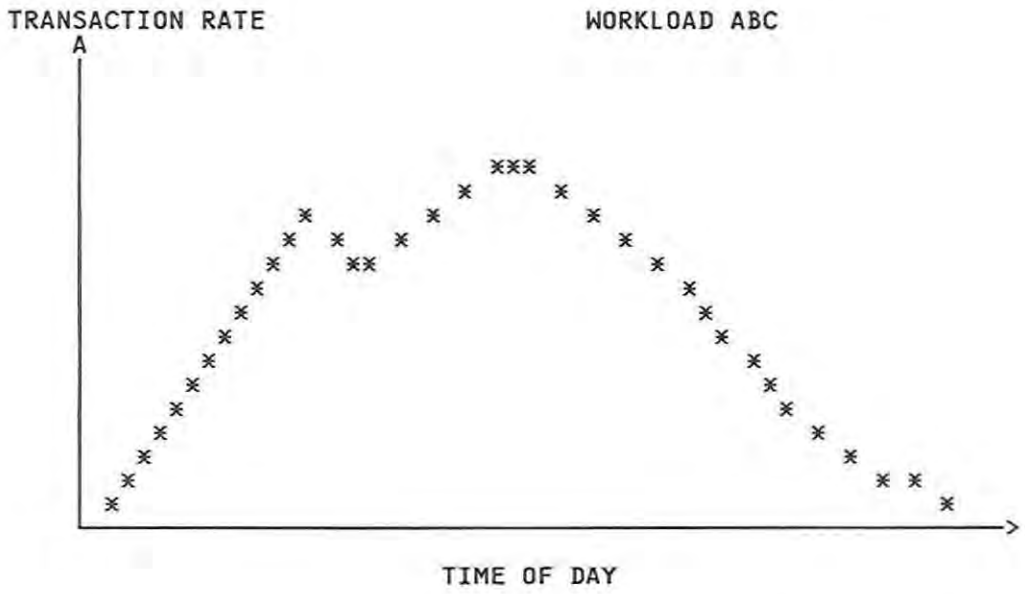
How do you quantify the response time requirement? One way is to find what the users think is correct. Another is to understand what the user does and evaluate what it takes to do a given amount of that per hour or per day. Then relate this information to the needs of the business. Some technical measurement data is needed -- in particular, transaction counts corresponding to user work.

Knowledge of the business and common sense may actually play a more important role in analyzing response time.

Response time estimate:



Next consider the transaction rate requirement. The transaction rate requirement for a workload is best built from experience (tracking). This transaction rate is a composite of the "average user" transaction rate and the concurrent users for this workload. The concurrency includes the number of users, number of terminals, time zones, user habits, the phase of a project, recent business activity (and business patterns), customer activity, ... For each workload, look for transaction rate patterns. There often is a time of day pattern. For program development, there may be time of day and phase of project patterns. In some cases there may be four or five patterns that make up the composite transaction rate. If these patterns can be found, the result is simpler analysis and a smaller "data base" for capacity planning data. With stable patterns we need only the patterns and an average value to describe the transaction rate requirements. Check the transaction rate data periodically to evaluate that the patterns have not changed -- or make adjustments accordingly. Transaction rate data should be tracked continuously. The amount of history that is kept depends on what the installation needs for making decisions.



From time to time transaction rate estimates are needed for new applications. There is no one way to proceed. Develop what works for you. Some guidelines may be useful. Start with what the end-user does and how he/she does it. Do a possible scenario of the activity and estimate transaction counts and patterns. For discussion, consider two cases. In case 1 the end-user reports his/her daily activity. That activity consists of about 78 (RMF) transactions to complete a business form. One business form is completed for each customer contacted by this user. Estimate a scenario in time of the user interacting with the customer to complete the form (78 transactions). Based on expected customer availability and user habits, estimate an activity pattern. Estimate the number of concurrent users. Then, with a little smoothing and fudging, we have a transaction rate requirement.

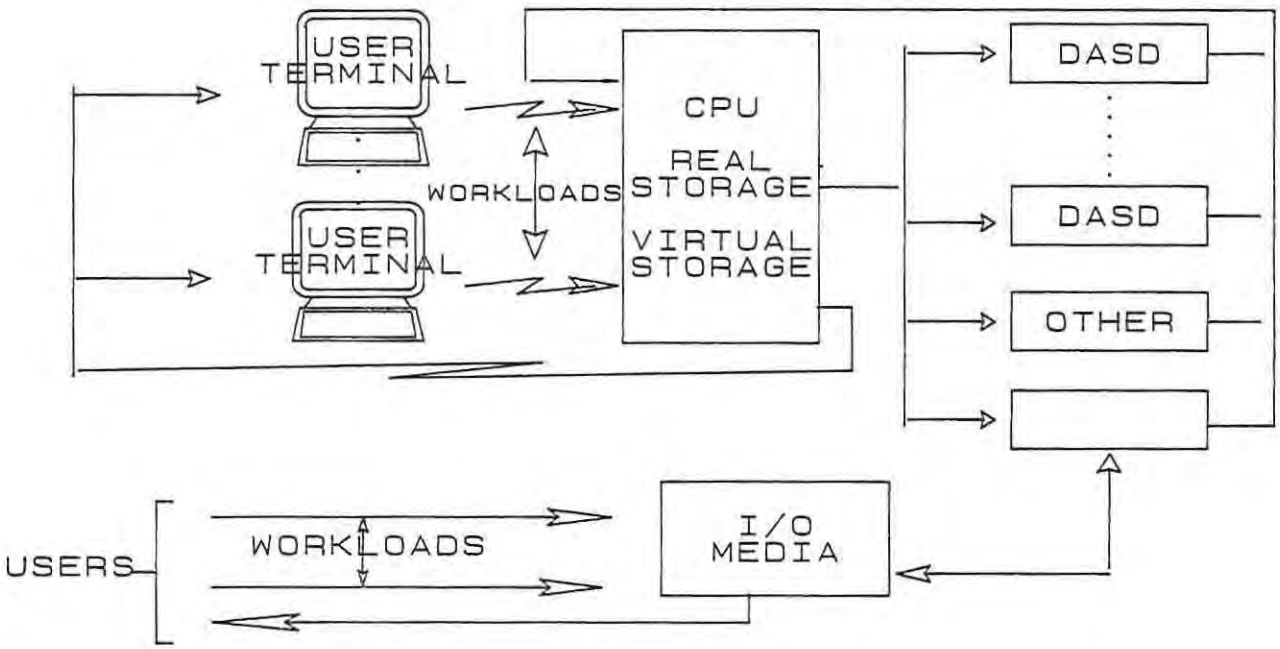
In Case 2 (a very different case) the end-user analyzes economic activity. In this case the user has subscribed to some data bases and has purchased an econometric model, so we do not get a chance to watch the activity start small and grow. Now the "boss" said that this activity has a certain (he gave a dollar amount) value to the business. Applying some cost-value relationships, we would typically expect this activity to support "X" amount of resource consumption. Therefore we set control parameters to provide this X amount of resource and proceed to watch and track very carefully. The user and the boss should know what we have done and why we are doing it. Notice something? This procedure is almost never found in DP environments (?). For normal, non-computer business decisions this kind of process is an everyday occurrence. Somehow, in a computer environment the process gets lost and DP planners get paranoid about how much resource this user might consume. Another guideline: you are allowed to limit (i.e., manage) resource consumption. Do it for good business reasons -- and check with the boss.

QUANTIFY THE DP SERVICE

So far we have been talking about planning and business considerations -- a general and broad topic area. Now we shall get into some hi-tech DP measurement techniques. This is the "other half" of preparation for capacity planning.

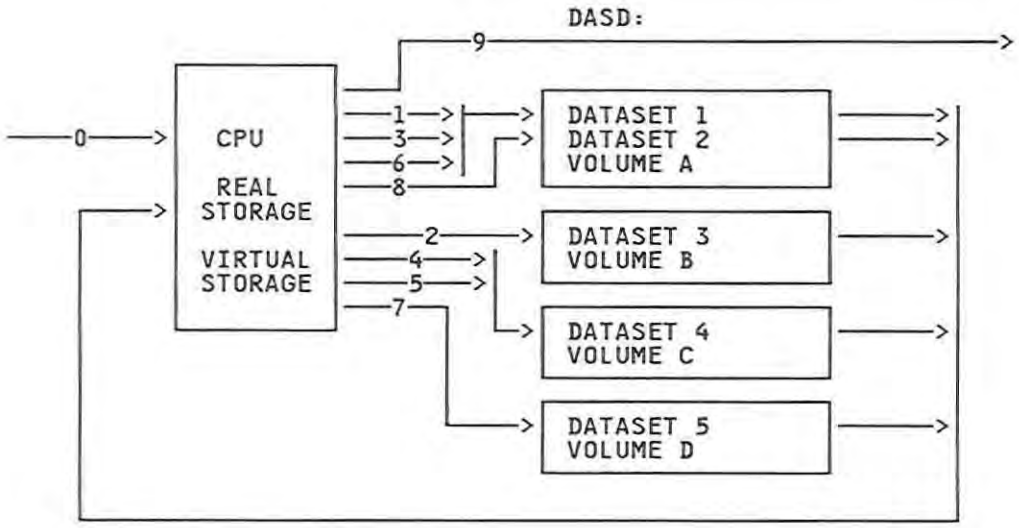
To understand performance (and capacity) in a DP system is to understand the transaction flow in that system. What is the maximum rate of transaction flow (or minimum time that a transaction takes)? What are reasonable and expected delays relative to this potential capacity? When is a delay too large? Some of the more sophisticated relations come from queueing theory. A lot of practical analysis can be done with utilization calculations. The fundamental ingredient for either queueing theory or utilization is service time. Or, more generally, service.

THE DP SYSTEM



What is "service"? Service is the resource consumption to do one "average" transaction within the transaction activity for a workload. Sounds too simple? Why do you want something complicated? Utilization -- the most used and abused measure of resource consumption -- is important and necessary in an analysis but knowing the components of utilization, service time and transaction rate, is key to understanding performance and capacity. For most cases, service is in units of time. Other units of resource consumption include frames or bytes for a real storage working set or a virtual storage requirement. For DASD it may be convenient to keep IO accesses per transaction and data bytes per IO as a measure of service and then convert this data to time in an analysis. Be sure to keep track of the units!

For a system, there are many service times. Consider a transaction flowing thru the system:



Each pass through each "server" accumulates some amount of service time. For the above illustration, there are 9 passes through the CPU, 4 IO's to volume A, 1 IO each to volumes B and D, and 2 IO's to volume C. In some cases a breakdown by dataset is desirable. Many installations use just the volume data. The service times for this transaction flow are a CPU service time representing the 9 passes through the CPU and and four (sets of) numbers representing the delays through each of the 4 volumes. Again, the DASD data might be kept as the number of IO's for each volume and a service time for each volume, or dataset, or ... (what do you need to make decisions?). For more detailed analysis, it is desirable to know the data length. Data length can be used to evaluate path utilization, RPS delay, and service time per IO. (Average latency is known and actuator seek time can be estimated.) In some cases locks are involved. Basically, locks translate into longer service times; the server is busy to everybody else while the lock is held.

Know your measurement and data reduction tools. List the variables that are needed for analysis in your environment and find a tool(s) that reports those variables. Then, if you can not buy it, code it -- unless you do not want to manage it. There are 3 distinct phases to the performance analysis process: data collection, data reduction, and data analysis. First is the data collection. Hopefully most of the needed data is collected by the programs and code necessary to run the system. In MVS, this includes the SRM data and RMF/SMF records. Subsystems, such as IMS and CICS, often provide additional data collection facilities. When system data is needed that is not collected by the system, a RYO program that collects that data should be considered. The good news is to consider this alternative and decide you do not need it. Such a program is likely to be difficult to do and difficult to maintain. Data collection code within an application is another story, of course. For applications, measurement and management techniques should be part of the application planning and requirements. Application data, when synchronized with system data can be very useful. Sometimes the number of IO accesses by this application to specific data sets is important, for example.

Next is the data reduction. What we need is a tabulation of service per transaction by workload for each resource.

In an MVS system, Performance Group Numbers are useful for structuring the data collection process to provide data by workload. The ICS member of PARMLIB can be used to assign Performance Group Numbers. This data collection is reflected in RMF Workload reports. SMF Type 30 records with Interval Accounting give some additional data on IO activity (but do not include data by Performance Group Period).

There are three categories of this data:

1. Resource data by workload,

2. Resource data by a support program such as JES or VTAM but without a breakdown by workload, and

3. Resource data without any connection to workload or any other program.

This third category is often called "uncaptured" time or service; uncaptured CPU time belongs to this third category. It would be nice if all data were assigned in some automatic way to a workload or a system function. However, it is not to be, and there are many challenges for the analyst to pursue. If possible, assign or apportion (unassigned) resource consumption to workloads by some variable that is part of the workload data. For example, assign VTAM to workloads in proportion to the number of transactions. Assign JES in terms of a combination of transactions and lines of print. If you have something better, use it. Chances are that some reasonable apportionment is much better than none. For resource consumption data without any connection to workload or program, consider a statistical approach. How does variation in this consumption correlate with the different workloads? Techniques of multiple linear regression may apply here.

One of the more difficult consumptions to deal with directly is the additional CPU time an MP spends looking for work. This might be modeled as a constant multiplied by the frequency of entering wait state. If it is ignored, it will show up as smaller capture ratios for the workloads at "middle" utilizations. The object is to find a way to account for all resource consumption. When this is accomplished, the workloads can be migrated or grown, as required by the business plans.

Much of the CPU resource data collection is by workload (or support program). There are, however, two topics that require more attention. They are (1) transaction count and (2) capture ratio. For TSO and batch workload-types, transaction counts are reported in RMF and CPU per transaction is easily calculated. For other workload-types, other reports may be required. For CICS, transaction count data can (optionally) be reported to RMF via the SYSEVENT interface. The SYSEVENT interface is available to any Subsystem. Capture ratio is is important in terms of understanding where the CPU resource went. CPU capture ratio (CR) is a measure of total time relative to TCB + SRB time. While CR is not a fixed number, it is "constant enough" to be useful. CR may vary from release to release of MVS. CR is likely to go down if the workload is moved from a UP to an MP (this appears to be because we ignore some additional MP kind of activity). When evaluating CR's, view the system as workload-types rather than workloads. This reduces the number of CR's to be calculated, and, therefore, the number of sets of data required. The number of sets of data must be at least as many as the capture ratios desired. Also, each set of data should be from a "representative" interval and the activity levels (for each workload-type) in each set of data must be in different proportions. Programs are available to help calculate CR's. If the system is dominated by one workload-type, consider calculating a "system-wide" CR. This is a simple calculation: (CR = ((TCB + SRB time)/(U * M * Interval))) from an RMF report. U is average utilization per engine. M is the number of engines. It only gets complicated when multiple workload-types are involved.

Back to service time. How much data is required to say that we have a valid average for planning purposes? There is no sure-fire answer. For a large TSO system, 2 hours of data from 1 to 3 on a normal Thursday afternoon may come rather close. For some workloads, several months of data may be needed with additional data for "year-end" considerations. The lower the workload structure, the more difficult the task of evaluating service time requirements. INFO Center, for example, is one of the more difficult cases. There may be some mathematical approaches to this issue, but the pragmatic one of tracking the data and looking for stability is probably the most benefit for the least cost.

For each resource in the system, collect data relative to how you want to model the resource and how you want to make decisions. In some cases consider a "divide and conquer" approach. For example, we could assume for capacity planning that the path utilization for the primary on-line workloads will be less than 20% and that these paths will not be shared. This is relatively easy to achieve and often necessary anyway. And the on-lines are worth it! This approach makes the calculations much simpler. There may be other simplifications that apply to a given environment -- specifically, your environment. Write down all the assumptions.

Note that for tape, a good resource to model is the Control Unit. The system performance bottlenecks (or lack thereof) are often at the CU. Decisions on number of drives may be separate and depend on the number of drives per job, job elapsed time, concurrent jobs, operator and tape library efficiency, and maintenance requirements. We probably don't want all that in the same model as queueing for the CPU, etc. The divide-and-conquer approach can be very effective, but the "divisions" must be carefully and properly chosen.

So far the discussion has been on service measurement data. What if the workload does not exist yet? There is no one answer here -- and do not expect perfection. A good way to start is with the non-paging IO's per transaction. This measure should have relevance to the application designer -- it should have logical value and meaning. What data does the transaction need and how many IO's is that? In some cases a physical IO is not required. Can that be estimated? Probably it can. Next, drawing upon "similar" workloads, estimate CPU per IO and a real storage working set. If real storage is managed, demand pages per transaction can be estimated. In some cases, estimating virtual storage is appropriate.

The result of the service time calculations is a service time table.

	WORKLOADS				NUMBER
RESOURCES	01	02	03	...	SERVERS
MAX WKLD CONCUR'CY	_____	_____	_____	_____	
CPU	_____	_____	_____	_____	_____
REAL STORAGE WS	_____	_____	_____	_____	
DPAGES/TRAN ...	_____	_____	_____	_____	
SWAPS/TRAN	_____	_____	_____	_____	
PAGES/SWAP	_____	_____	_____	_____	
VOLSERxx	_____	_____	_____	_____	- - - -
...					
VOLSERxx	_____	_____	_____	_____	- - - -
PATHyy	_____	_____	_____	_____	_____
COMMUNICATIONS	_____	_____	_____	_____	
USER THINK TIME	_____	_____	_____	_____	

CAPACITY PLANNING

With a thorough job done on (1) the business plan and requirements and (2) the DP service, the rest is rather easy. If we skimp on the first two, the rest can be never ending turmoil and confusion. At this point the business requirements have been quantified:

BUSINESS ELEMENTS	**RESPONSE TIME**	**PRIORITY**
WORKLOADS	**TRANSACTION RATE**	**AVAILABILITY**
		FORECAST

The DP service has been quantified as a SERVICE TIME TABLE.

Now on to capacity planning. Capacity planning topics include:

SHARED-SPACE	**AVAILABILITY**
TRANSACTION FLOW	**CONTINGENCIES**

In some cases the names of variables are the same; however, the variables are different. The business requirement response time is an objective and requirement. The implementation design response time is planned attainment (calculation). The actual response time (tracking) is the attainment.

Next we need to quantify the resource requirements for running one or more workloads in each (installed or planned) system. Two specific areas of interest are (1) shared-space resources and (2) the resource consumption associated with transaction flow. Shared-space resources include

REAL STORAGE, VIRTUAL STORAGE, and DASD SPACE.

The CPU, paths, and control units and/or devices are shared-time resources and are addressed in a transaction flow analysis. Transaction flow analysis includes

UTILIZATION, CONCURRENCY (MPL or QUEUE), and RESPONSE TIME

for these resources. (Where the response time attained is to be equal to or better than the business requirement response time.) Shared resources must be managed as a whole. Managing or tuning one piece of a shared resource will probably cause trouble somewhere else unless the whole picture is considered. For this reason, calculate and track a map of real storage for each system. See Appendix A1, First, there is a resident storage portion including the Nucleus, SQA, LPA, and CSA. Allow some frames for the Available Frame Queue (AFQ). The AFQ allows a swap-in (or similar action) to occur without waiting for pages to be paged-out. The result is a more responsive system. Allow some frames for the logical swap queue. As a minimum, this is the number of frames to be available for logical swap as real storage becomes "full". Since logical swap is more efficient than physical swap, we do not want it to go away just when it is needed the most.

This can not be controlled in a precise way, but the Minimum System Think time parameter in the OPT member of PARMLIB does provide some control. A non-zero value will allow some logical swap when real storage is full.

Next allow some frames for the non-swap workloads. Allow a number of frames that results in satisfactory response times for the particular end-users. For example, a test CICS might have more demand pages per transaction than a production CICS but each would have a satisfactory (but

different) response time. Note that demand pages per transaction is the most sensitive variable for evaluating working set size. This is not as easy to see in the measurement data as before because trim algorithms in current releases of MVS allow real storage to distribute in proportion to need. Some of the older algorithms trimmed just to be tidy. Consider using Storage Isolation for non-swap ASIDs. Do not do Storage Isolation without a real storage map. Last, allow some frames for each Domain with swappable ASIDs. Calculate the working set size (WSS) for each workload. The WSS together with the concurrency (MPL) required for performance determine the real storage requirements for the swappable workloads. In general, do not mix swappable ASIDs and non-swap ASIDs in the same Domain.

Track virtual storage usage. Maintain an analysis of virtual storage. Focus on LPA, CSA, and major users of Private. CSA Storage Protect Key 0 is the most difficult part. Be sure to track before and after major changes. Based on tracking and announced product requirements, estimate future VS requirements. Take advantage of 31-bit products wherever possible. Maintain a VS map(s) for each system. (Appendix A2.)

Most of the attention on DASD goes to IO rate and response time. Capacity planning should include an evaluation of DASD space, as well. A DASD space map can be used to evaluate isolation by volume and by path. (Isolation means -- as a minimum -- keeping response oriented activity away from non-response oriented activity.) Isolation is a key technique in achieving consistent performance. There are seven basic kinds of DASD space to manage. See Appendix A3.

SYSTEM	**DATABASE DATA**	**DFHSM LEVEL 0**
PAGE DATASET	**DATABASE PROGRAMS**	**DFHSM LEVEL 1 (N-1)**
SWAP DATASET		

The database volumes need to be managed to the requirements of the database applications. However, general TSO and batch requirements can be managed by DFHSM. From a system point of view, we now have something to manage. No longer are we stuck with the emotional choice of providing everything requested or risking disaster somewhere down the road. We can plan an amount of Level 0, monitor the activity, and develop a cost effective system. In time we might even come up with some theories. We can manage performance and do not risk losing programs or data. In fact, the programs and data are safer. Find a way to keep track of DASD space; include the (logical) paths so that isolation alternatives can be evaluated.

Next lets look at some transaction flow analysis. Utilization analysis is the most used technique in capacity evaluations. It often gives the most benefit for the least effort. Utilizations can be used together with "Rules Of Thumb" (ROT) or can be a first step in a modeling process. For the ROT process consider:

WORKLOAD TYPE	RESOURCE	MAX. UTILIZATION
ON-LINE	CPU	50% - 80%
BATCH	CPU	80% - 100%
ON-LINE	DASD	30%
BATCH	DASD	50% - 90%
ON-LINE	DASD PATH	20%
BATCH	DASD PATH	30% - 50%

The basis for selecting these percentages is a general estimate that response time is still less than two to three times the service time. ROT are easy to use, can give the wrong answer, and are no longer in-vogue with out-of-town experts. Modeling to evaluate response time is the best answer but requires more effort and skill. Fancy models are also capable of giving wrong answers. Consider a combination of the two methods: a utilization analysis to know what may be possible (and what is "in-bounds") and a response time model for fine tuning. Again, what do you need to know to make decisions?

Here is a utilization model you could probably put on a spreadsheet:

TRANSACTION RATE

WORKLOAD —>	1	2	3	. . .
TRAN RATE	A	B		

SERVICE TIME TABLE

WORKLOAD —>	1	2	3	. . .
RESOURCE 1				
RESOURCE 2	X	Y		
RESOURCE 3				

UTILIZATION TABLE

WORKLOAD —>	1	2	3	. . .	UTILIZ.
RESOURCE 1					
RESOURCE 2	A*X	B*Y			(A*X)+(B*Y)
RESOURCE 3					

Then, with a response time model, add the following table:

RESPONSE TIME TABLE

WORKLOAD —>	1	2	3	. . .
RESOURCE 1	J			
RESOURCE 2	K			
RESOURCE 3	L			
. . . SUM ——>	J+K+L			

A concurrency (queue size) table could be generated from Little's Law: (response time) = (queue size)*(interarrival time). The interarrival time is (1/transaction rate) and queue size includes the transaction in service.

QUEUE SIZE TABLE

WORKLOAD —>	1	2	3	. . .	SUM QUEUE
RESOURCE 1	J * A				
RESOURCE 2	K * A				(———>SUM)
RESOURCE 3	L * A				
... SUM———>	(V SUM)				

The above tables represent a capacity planning design technique. For example, change or remove a transaction rate and recalculate the tables. Similarly, consider a different IO device, reevaluate the service time(s) and recalculate the tables.

A tracking chart(s) should be tabulated to go with the system design charts. Tracking is important. Tracking shows how the business elements are performing and indicates when further action should be taken. Tracking can be educational. Tracking should be 'top-down' -- that is, daily tracking should show the overall condition of the business elements. Then more detailed data should be available, if required. Track transaction rate and response time continually. These variables reflect why we (the DP installation) are in business. Track the Service Time Table as required. The Service Time Table should not change significantly unless we do something to cause the change.

Note that if two independent workloads are evaluated as one (combined) workload, the resulting average service time(s) will not be stable and will reflect the proportions of the two individual workload activity levels.

Consider the following tracking chart.

WORKLOAD NAME ———————— SYSTEM ———— DATE ————

TIME OF DAY	NUMBER CONCUR'T USERS	TRAN RATE (UNITS)	RESPONSE TIME (UNITS)	WORKING SET SIZE	DEMAND PAGES PER TRAN	NON-PAGE IO'S PER TRAN	BUSINESS ELEMENT DRIVER
V							

- If the transaction rate and response time are OK, perhaps we may take the afternoon off.
- If the transaction rate is high or low, check the number of concurrent users first.
- If response time is high, demand pages per transaction and real storage working set size are a good place to look first. Tracking these variables will teach us what size working set is satisfactory.
- The non-page IO's per transaction indicates whether or not the work itself has changed.
- Business element drivers should be tracked -- we depend on them for analyzing future requirements.

The most likely areas to inhibit attaining planned transaction rates or response times are the shared resources. Maybe this is because some workloads tend to be forgotten or ignored when the shared resource is analyzed. Have you ever heard, "... oh, that doesn't happen in our system ..."? The first

step, from a design point of view, is to isolate parts of the resource to each workload. In other words, create the appearance of an un-shared resource.

CPU	Dispatching Priority
Real Storage	Storage Isolation (non-swap ASID)
	Domain TMPL (swappable ASID)
DASD	by Volume
	by Path
Subsystem ..	Concurrency (high)
Locks	A structure with enough locks and short locks

A next step is to be sure that all the sharing workloads for this resource are included in the analysis. Note that a subsystem or ASID (for example, CICS) can act like a shared resource. From a performance view, watch out for resources or facilities that limit concurrency. Expect trade-offs to occur between performance and the data integrity that goes with a concurrency of one. For example, a subsystem that processes only one transaction at a time creates, in effect, a long "service time" for that ASID. The subsystem service time is the sum of the CPU and IO response times for a transaction. Lets take a best case.

This subsystem has top priority for the CPU and does not share IO. Assume an average transaction is 12 IOs with 2 milliseconds of CPU per IO. Each IO takes 20 ms (average). Then a transaction has a service time of 264 ms relative to this ASID. The maximum transaction rate for the ASID is 1/0.264 or less than 4 transactions per second. With no other workloads present, the maximum CPU utilization is 0.024/0.264 or 9%.

Include all the workloads that share a resource when analyzing a shared resource. This may sound simple but is often overlooked. Then manage and control the resource to workload objectives. In this way the resource should produce the delays that were modeled. Model a managed environment; "fix" an unmanaged environment. If you bought an automobile with no accelerator, no brake, or no steering wheel, would you (a) drive it, (b) model it, or (c) fix it?

Another system interaction that can cause difficulty is when different programs acquire locks in opposite sequences. This may not be capacity planning, per se, but it can reduce the capacity in a hurry. A total solution may not exist, but each installation should develop conventions for the sequence of requesting locks (exclusive enqueue) to the extent that this is practical.

Availability means that some workloads can run on one part of the configuration while another part of the configuration is down. Availability requires capacity considerations. Either there are deferrable workloads or there is redundant capacity. In both cases there is more connectivity in the configuration than otherwise. Availability starts with a definition of what workloads must be "up" during what time intervals and -- in the complete case -- at what cost. Note that this definition potentially includes conflict and trade-offs. The definitions are driven by value to the business. "Must be up" is not an absolute term but implies numbers like 98% or 99% available within a specified window.

AVAILABILITY

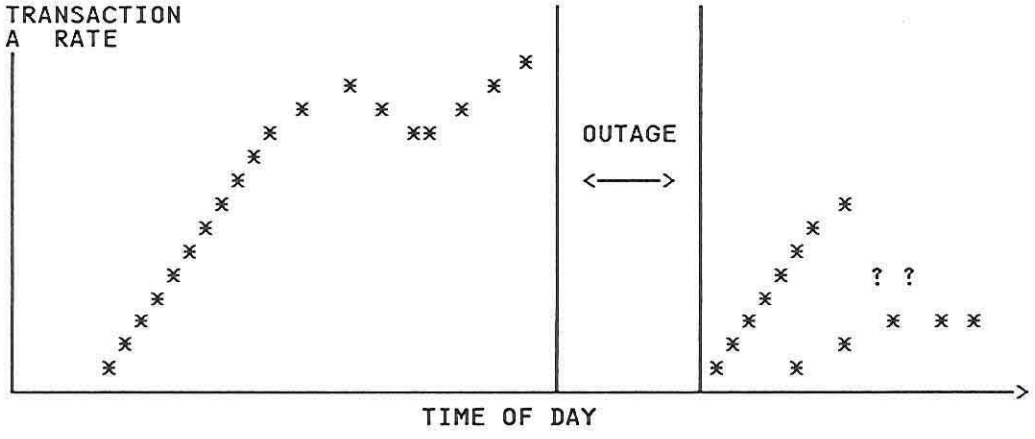
MVS SUBSYSTEM(S) APPLICATION(S) CPU REAL STORAGE VIRTUAL STORAGE DASD I/O CONFIGURATION NETWORK	INTERVALS: PRIME TIME SCHED TIME UNSCHED TIME OUTAGES: FREQUENCY DURATION TOTAL TIME

END-USER	END-USER SENSITIVITY

Attainable availability depends on configuration redundancy relative to the designated high availability workloads, configuration connectivity, application design, recovery procedures, and testing. Application design affects the number of system components that must be available at the same time and the reliability of those components. At times of major configuration change, a Component Failure Impact Analysis (CFIA) should be performed. A System Outage Analysis (SOA) should be an on-going analysis. The CFIA will help identify requirements for configuration redundancy and application redesign. The SOA will help identify system components that should be better stabilized or otherwise avoided for critical workloads.

Availability and recovery include addressing critical workloads, redundant configuration, rerun capacity, and delayed schedules. The critical workloads and redundant configuration can be evaluated as a subset of the transaction flow analysis system design (above).

Rerun capacity is often "built-in" to the way workload data is collected. If not, some attention must be given to rerun requirements. Note that a major shift in availability or reliability could have an impact on rerun requirements. Delayed schedules due to outages are an elusive item to quantify. They are somewhat easier to quantify when associated with end-of-month closing or an overnight batch window. Another capacity factor on prime shift is the procedure to notify users that the system is now available. How is this done at your installation? How quickly does the load return to normal values after the system becomes available?



With a business element structure of the workloads, you are in a position to evaluate contingencies. How much contingency is required for "payroll"? How much contingency is appropriate for seismic analysis? Where is the opportunity?

If all the workloads peak at the same time, do you want to maintain response time levels? Are there special considerations for month-end activity? These questions have answers when they are applied to an environment and to a set of objectives. The capacity can be quantified because of the business element structure of the data.

Most of the time, capacity planning considerations apply to a given overall system design and set of "real estate". Once in a while, there is a broader scope. What then? What are the real constraints? There is probably more than one answer or direction. Your business requirements may dictate unique approaches. However there are two basic considerations:

1. What business data is required and how can it be distributed?
   What are the "common database" considerations?

2. Where are the end-users and
   What options will work as end-user locations?

Computers and networks can be configured in many ways. End-users and common business data are often the real constraints.

Will we grow forever? Are there limits? Consider the following four areas:

(1) TRANSACTION RATE:
EMPLOYEES ON-LINE.
SUPPLIERS ON-LINE.
CUSTOMERS ON-LINE.
BUSINESS FUNCTION ON-LINE.

(2) SERVICE TIME:
FUNCTION/TRANSACTION.
COLOR GRAPHICS.
USER FRIENDLY.

(3) APPROPRIATE DATA ON-LINE:
THIS BUSINESS.
THIS COMPANY.
SUPPLIERS.
CUSTOMERS.

(4) SYSTEM CONSIDERATIONS:
ON-LINE UPDATE.
HIGH AVAILABILITY.
EXPERT SYSTEMS.

Some capacity planning thoughts:

IF YOU DO NOT HAVE PERFORMANCE CRITERIA, YOU ARE NEVER OUT OF CAPACITY.

THE "SYSTEM" IS NOT OUT OF CAPACITY, "BUSINESS ELEMENTS" ARE OUT OF CAPACITY.

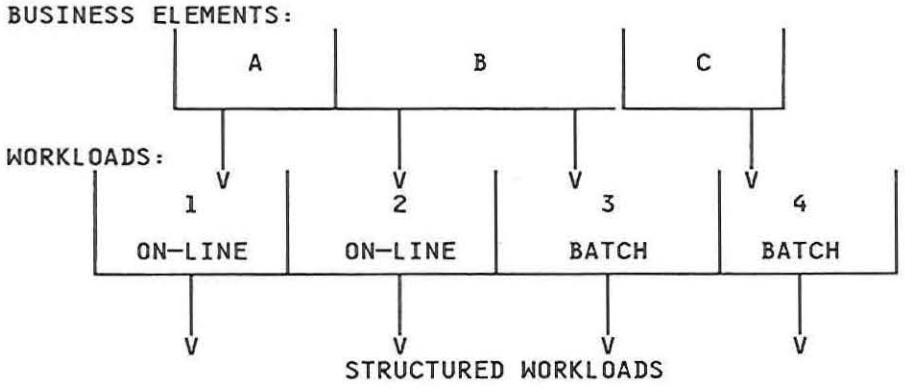
MVS CONTROLS

MVS controls play a roll in capacity. Use the MVS controls (principally the SRM) to help manage the system. The SRM does a good job but does a better job if you give it some guidance. In particular, three PARMLIB members communicate your guidance to the SRM: ICS, IPS, and OPT. The SRM is most significant when the system is busy. Consider the following analogy to a bank at noon on Friday: It really does not matter whether there is a common queue for all tellers or individual queues for each teller as long as there are more tellers than customers.

And so with the SRM, the parameters have little effect when the system is lightly loaded but can be essential at heavy loads.

Performance Group Numbers (PGNs) allow a system structure that can be tracked and managed. Assigning PGNs is a key step in the whole process. This structure is the link between what is important to the installation and the otherwise expressionless 1's and 0's.

PGNs can be assigned using the ICS. Then the PGNs can be controlled using the IPS. Paging, a major subsystem consideration, can be controlled using the IPS (Storage Isolation and TMPL) and the OPT (paging rate thresholds). For TSO workloads, logical swapping can be managed (to some extent) using OPT parameters. PGN assignments facilitate tracking through RMF Workload reports.



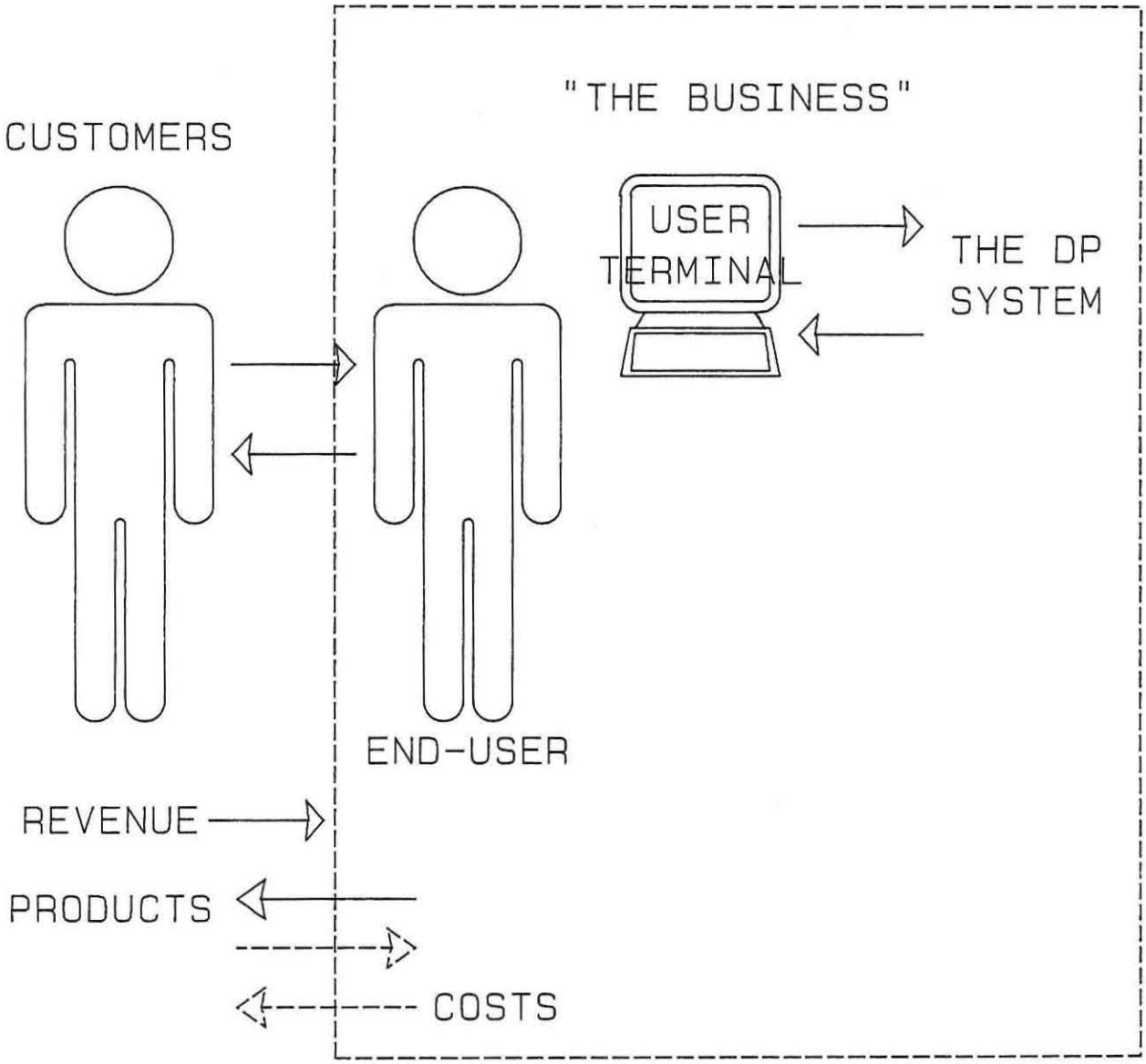
SUMMARY

Capacity planning covers a broad set of topics from business objectives to control parameters in the DP system. Through this process it is important to maintain a view of the objectives. Business objectives are the common thread that holds the whole process together. Then, having done such a wonderful job of data organization, measurement, and control, what do we tell the boss? How are performance and capacity reported to management? The logical answer at this point is to report in terms that relate to the business objectives. That means report business volumes and employee productivity.

BUSINESS VOLUMES	EMPLOYEE PRODUCTIVITY
- NUMBER OF TRANSACTIONS	- RESPONSE TIME
- NUMBER OF USERS	- PROJECT SCHEDULE
- NUMBER OF CONCURRENT USERS	- USER THINK TIME
- WHAT WORKS FOR YOU? (SPECIFIC TO YOUR BUSINESS)	- END-PRODUCTS/EMPLOYEES/TIME
- BUSINESS ELEMENT DRIVER	- WHAT WORKS FOR YOU?

AVOID INTERNAL DP PARAMETERS SUCH AS UTILIZATION !

TODAY'S SYSTEMS



Appendix A.

A1. Real Storage Map.

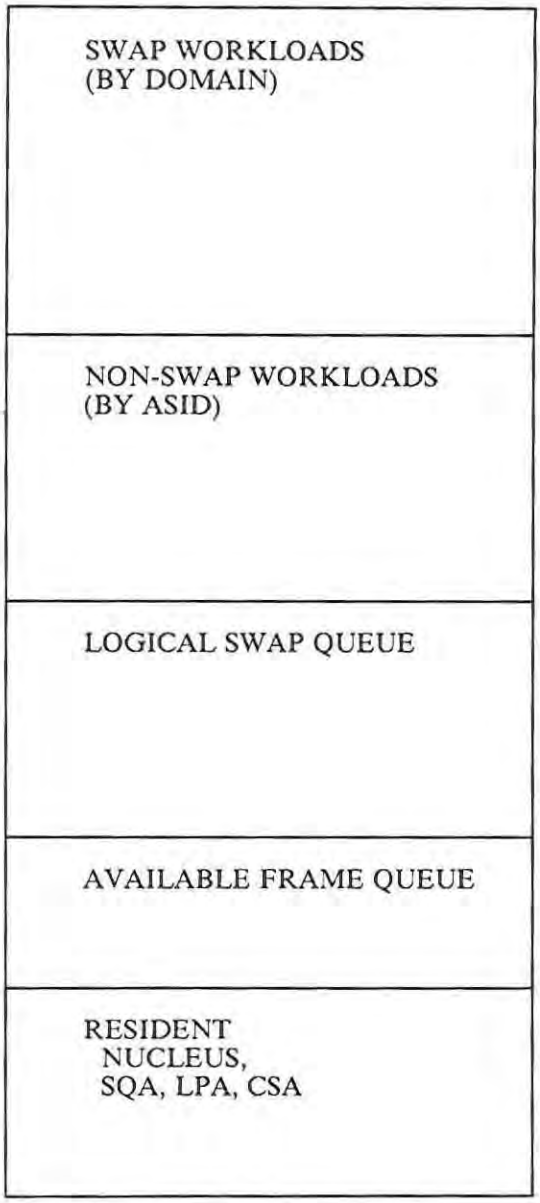
A2. Virtual Storage Map.

A3. DASD Space Map.

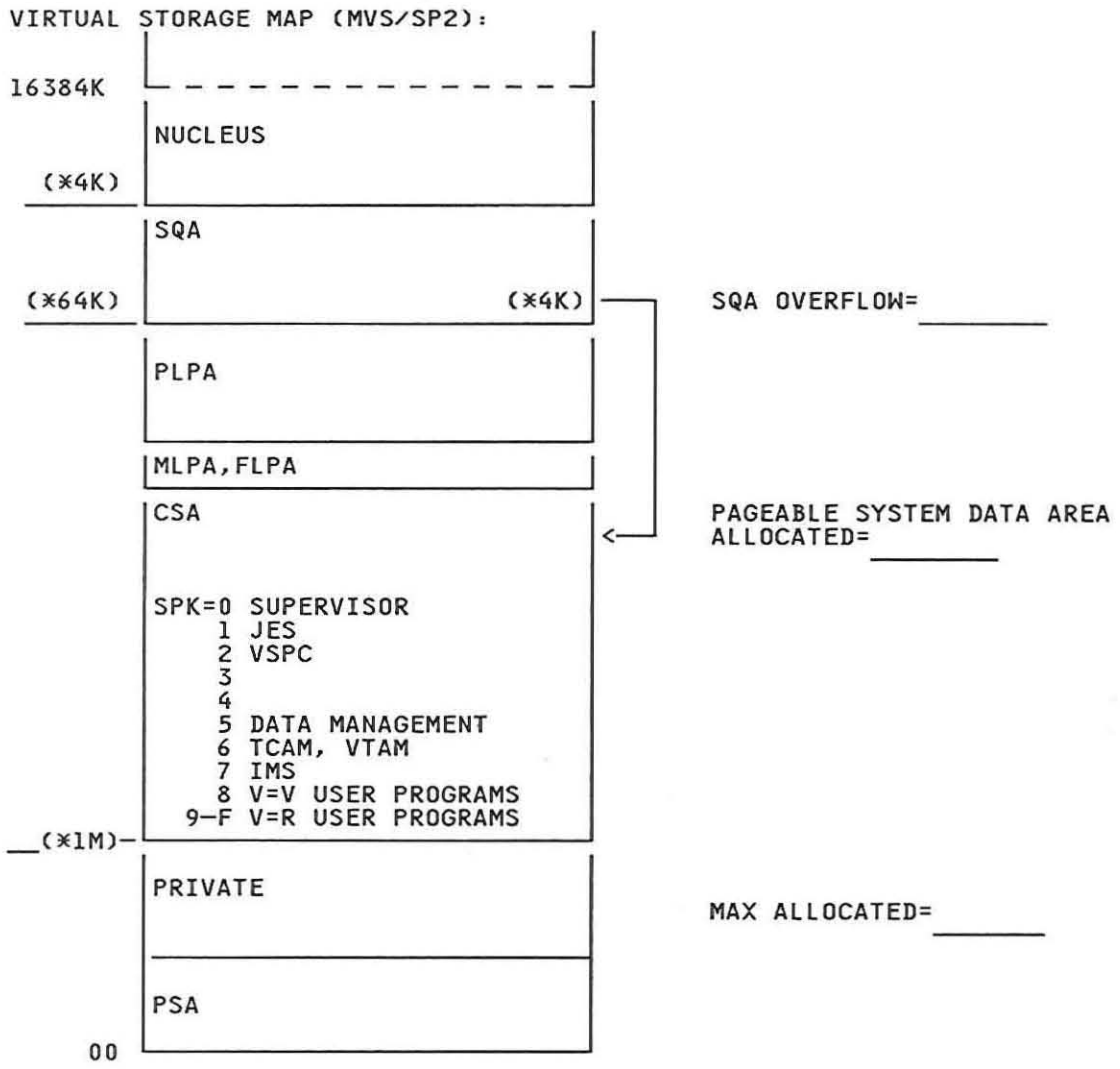
B1. Related and Supplementary Publications.

Appendix A1.

REAL STORAGE MAP:

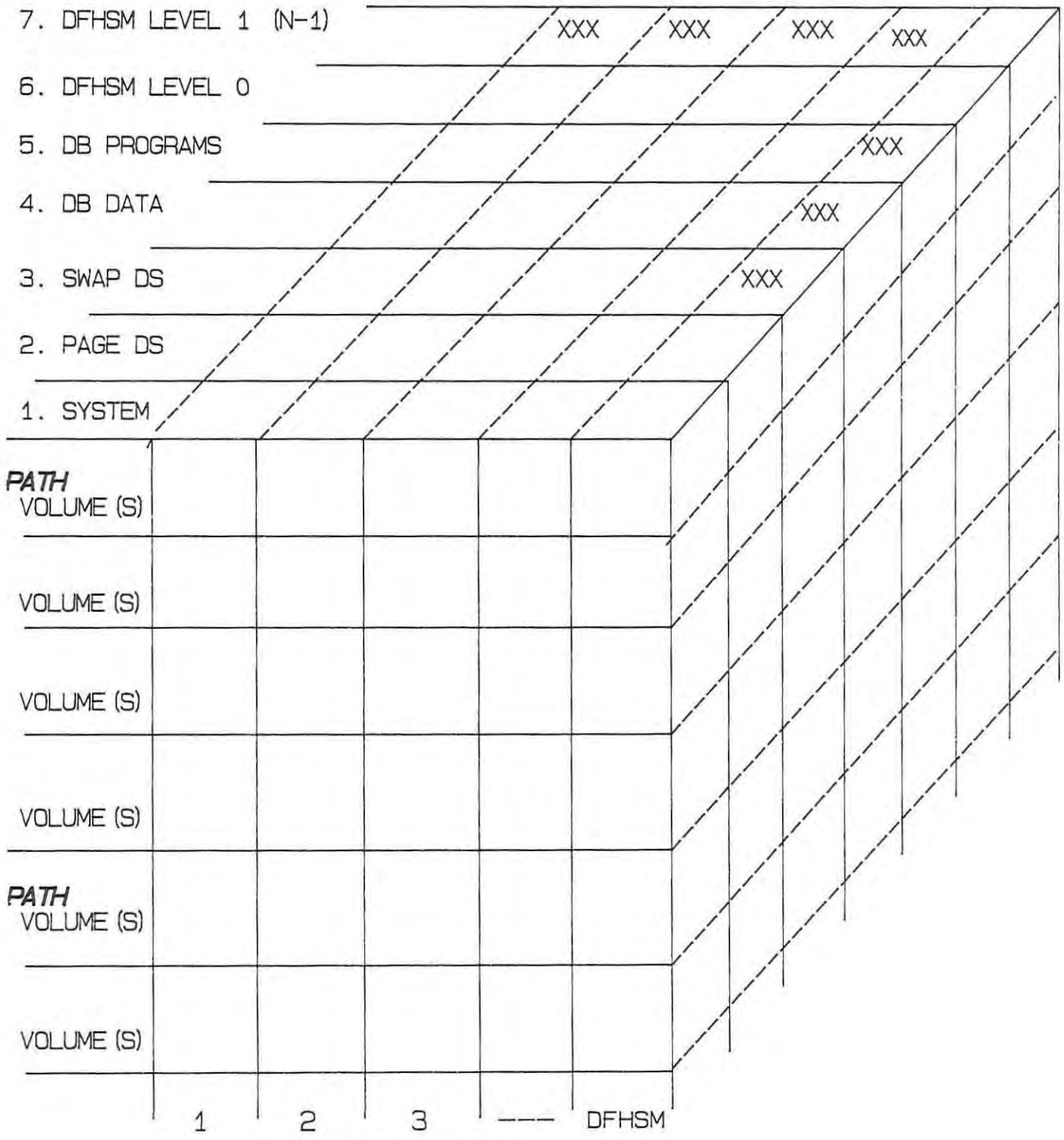


Appendix A2.


VIRTUAL STORAGE MAP (MVS/SP2):
16384K
NUCLEUS
(*4K)
SQA
(*64K)
(*4K)
SQA OVERFLOW=
PLPA
MLPA,FLPA
CSA
PAGEABLE SYSTEM DATA AREA
ALLOCATED=
SPK=0 SUPERVISOR
1 JES
2 VSPC
3
4
5 DATA MANAGEMENT
6 TCAM, VTAM
7 IMS
8 V=V USER PROGRAMS
9-F V=R USER PROGRAMS
(*1M)
PRIVATE
MAX ALLOCATED=
PSA
00

Appendix A3.

DASD SPACE MAP:



NOTE: 'PATH' MAY REPRESENT MULTIPLE PHYSICAL PATHS.

Appendix B. RELATED AND SUPPLEMENTARY PUBLICATIONS

Capacity Planning and Performance Management Methodology. GG22-9288-0.

DASD Access Method Considerations. GG22-9241-0.

MVS Performance Management. GG22-9351-0.

DASD Expectations. GG22-9363-2.

An MVS SRM Discussion. GG66-0201-0.

Capacity Planning Basic Hand Analysis. GG22-9344.

Analysis of Some Queueing Models. GF20-0007-1.

SLR Version 2 User's Guide. SH19-6215.

Capacity Planning Extended (CPX). SB21-2392.

SMF. GC28-1153.

RMF Reference and User's Guide. LC28-1138.

Initialization and Tuning Guide. GC28-1149.

System Outage Analysis. GC20-1871.

Component Failure Impact Analysis. GC20-1865.

Queueing Systems, Volume 1. by Leonard Kleinrock. John Wiley & Sons.

Probability, Statistics, and Queueing Theory. by Arnold Allen. Academic Press.

Statistics, An Introduction. by D. A. S. Fraser. John Wiley & Sons.

READER'S COMMENT FORM

Title: Capacity Planning Overview
Washington Systems Center
Technical Bulletin GG66-0254-00

You may use this form to communicate your comments about this publication, its organization, or subject matter, with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Please state your occupation: _____

Comments:

Please mail to:

R. M. Armstrong
IBM Corporation
Washington Systems Center
18100 Frederick Pike
Gaithersburg, MD 20879

Reader's Comment Form

Cut or Fold Along Line

Fold and tape | Please Do Not Staple | Fold and tape

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 40 ARMONK, N.Y.

POSTAGE WILL BE PAID BY ADDRESSEE:

R. M. Armstrong
Washington Systems Center
IBM Corporation
18100 Frederick Pike
Gaithersburg, MD 20879

Fold and tape | Please Do Not Staple | Fold and tape

IBM

GG66-0254-00

Capacity Planning Overview

Printed in U.S.A.

GG66-0254-00

IBM