CAPACITY PLANNING FOR THE DATA WAREHOUSE

BY

W. H. Inmon

CAPACITY PLANNING FOR THE DATA WAREHOUSE

The data warehouse environment - like all other computer environments - requires hardware resources. Given the volume of data and the type of processing that goes against the data, the data warehouse is capable of consuming large amounts of resources. For organizations that want to be in a proactive stance - where hardware resource utilization is not a surprise and the response time of a system is anticipated ahead of building the system, capacity planning for the data warehouse environment is a very important exercise.

There are several aspects to the data warehouse environment that make capacity planning for the data warehouse a unique exercise. The first factor is that the workload for the data warehouse environment is very variable. In many ways trying to anticipate the DSS workload requires imagination. Unlike the operational workload that has an air of regularity to it, the data warehouse DSS workload is much less predictable. This factor, in and of itself, makes capacity planning for the data warehouse a chancy exercise.

A second factor making capacity planning for the data warehouse a risky business is that the data warehouse normally entails much more data than was ever encountered in the operational environment. The amount of data found in the data warehouse is directly related to the design of the data warehouse environment. The designer determines the granularity of data that in turn determines how much data there will be in the warehouse. The finer the degree of granularity, the more data there is. The coarser the degree of granularity, the less data there is. And the volume of data not only affects the actual disk storage required, but the volume of data affects the machine resources required to manipulate the data. In very few environments is the capacity of a system so closely linked to the design of the system.

A third factor making capacity planning for the data warehouse environment a nontraditional exercise is that the data warehouse environment and the operational environments do not mix under the stress of a workload of any size at all. This imbalance of environments must be understood by all parties involved - the capacity planner, the systems programmer, management, and the designer of the data warehouse environment.

Consider the patterns of hardware utilization as shown by Figure 1.

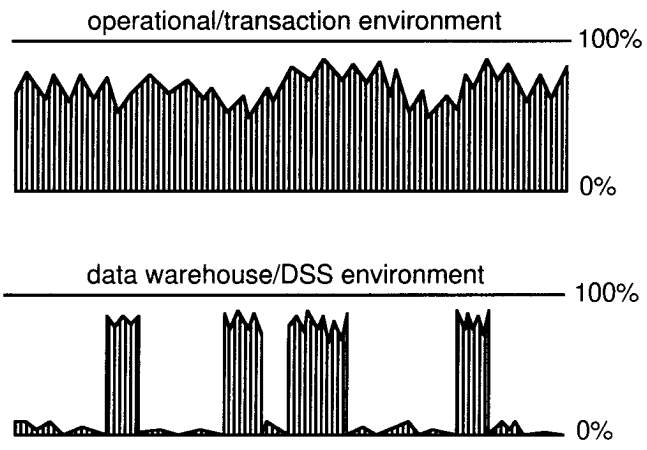


Figure 1: The fundamentally different patterns of hardware utilization between the data warehouse environment and the operational environment.

In Figure 1 it is seen that the operational environment uses hardware in a static fashion. There are peaks and valleys in the operational environment, but at the end of the day hardware utilization is predictable and fairly constant. Contrast the pattern of hardware utilization found in the operational environment with the hardware utilization found in the data warehouse/DSS environment.

In the data warehouse, hardware is used in a binary fashion. Either the hardware is being used constantly or the hardware is not being used at all. Furthermore, the pattern is such that it is unpredictable. One day much processing occurs at 8:30 am. The next day the bulk of processing occurs at 11:15 am, and so forth.

There are then, very different and incompatible patterns of hardware utilization associated with the operational and the data warehouse environment. These patterns apply to all types of hardware - CPU, channels, memory, disk storage, etc.

Trying to mix the different patterns of hardware leads to some basic difficulties. Figure 2 shows what happens when the two types of patterns of utilization are mixed in the same machine at the same time.

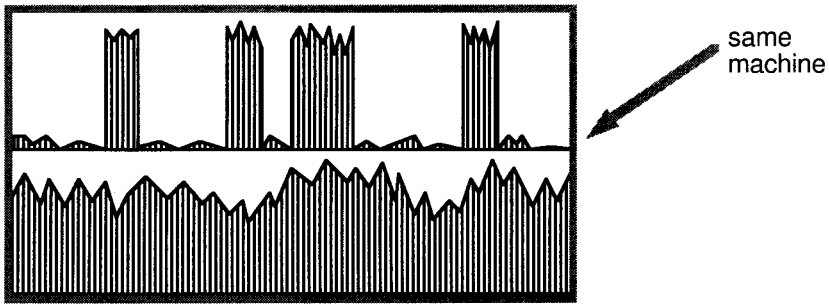


Figure 2: Trying to mix the two fundamentally different patterns of execution in the same machine at the same time leads to some very basic conflicts.

The patterns are simply incompatible. Either you get good response time and a low rate of machine utilization (at which point the financial manager is unhappy), or you get high machine utilization and poor response time (at which point the user is unhappy.)

The need to split the two environments is important to the data warehouse capacity planner because the capacity planner needs to be aware of circumstances in which the patterns of access are mixed. In other words, when doing capacity planning, there is a need to separate the two environments. Trying to do capacity planning for a machine or complex of machines where there is a mixing of the two environments is a nonsensical task.

Despite these difficulties with capacity planning, planning for machine resources in the data warehouse environment is a worthwhile endeavor.

TIME HORIZONS

As a rule there are two time horizons the capacity planner should aim for - the one year time horizon and the five year time horizon. Figure 3 shows these time horizons.

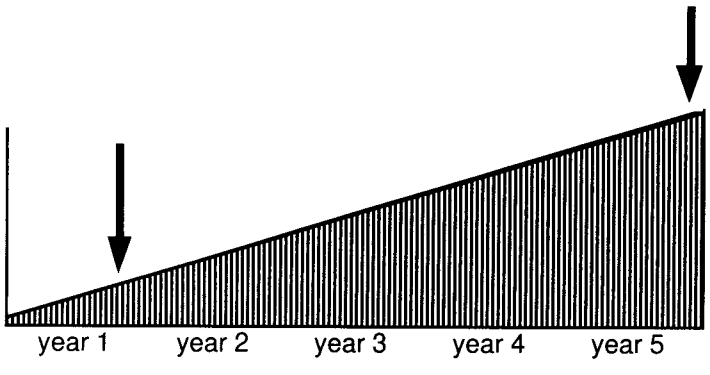


Figure 3: Projecting hardware requirements out to year 1 and year 5.

The one-year time horizon is important in that it is on the immediate requirements list for the designer. In other words, at the rate that the data warehouse becomes designed and populated, the decisions made about resources for the one year time horizon will have to be lived with. Hardware, and possibly software acquisitions will have to be made. A certain amount of "burn-in" will have to be tolerated. A learning curve for all parties will have to be survived, all on the one-year horizon.

The five-year horizon is of importance as well. It is where the massive volume of data will show up. And it is where the maturity of the data warehouse will occur.

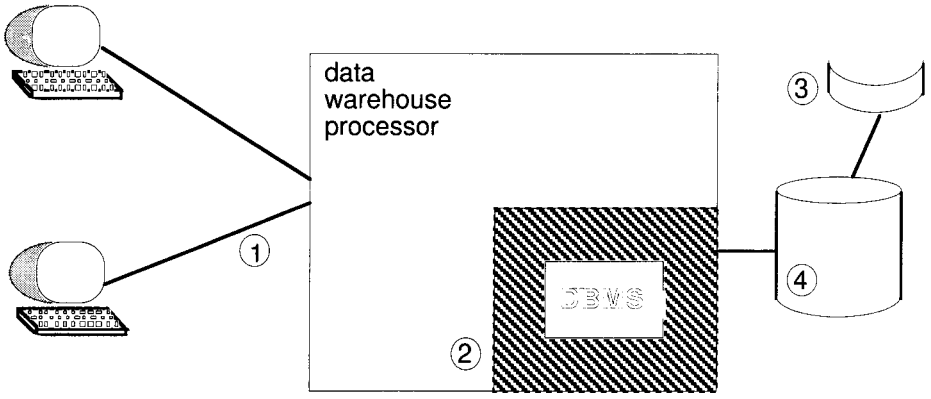
An interesting question is - " why not look at the ten year horizon as well?" Certainly projections can be made to the ten-year horizon. However, those projections are not usually made because:

- it is very difficult to predict what the world will look like ten years from now,
- it is assumed that the organization will have much more experience handling data warehouses five years in the future, so that design and data management will not pose the same problems they do in the early days of the warehouse, and
- it is assumed that there will be technological advances that will change the considerations of building and managing a data warehouse environment.

DBMS CONSIDERATIONS

One major factor affecting the data warehouse capacity planning is what portion of the data warehouse will be managed on disk storage and what portion will be managed on alternative storage. This very important distinction must be made, at least in broad terms, prior to the commencement of the data warehouse capacity planning effort.

Once the distinction is made, the next consideration is that of the technology underlying the data warehouse. The most interesting underlying technology is that of the data base management system - dbms. The components of the dbms that are of interest to the capacity planner are shown in Figure 4.



data warehouse DBMS capacity issues:
1. access to the warehouse
2. DBMS operations and efficiency
3. indexing to the warehouse
4. storage efficiency and requirements

Figure 4

The capacity planner is interested in the access to the data warehouse, the dbms capabilities and efficiencies, the indexing of the data warehouse, and the efficiency and operations of storage. Each of these aspects plays a large role in the throughput and operations of the data warehouse.

Some of the relevant issues in regards to the data warehouse data base management system are:

- how much data can the dbms handle? (NOTE: There is always a discrepancy between the theoretical limits of the volume of data handled by a data base management system and the practical limits.)
- how can the data be stored? compressed? indexed? encoded? how are null values handled?
- can locking be suppressed?
- can requests be monitored and suppressed based on resource utilization?
- can data be physically denormalized?
- what support is there for metadata as needed in the data warehouse? and so forth.

Of course the operating system and any teleprocessing monitoring must be factored in as well.

DISK STORAGE AND PROCESSING RESOURCES

The two most important parameters of capacity management are the measurement of disk storage and processing resources. Figure 5 shows those resources.

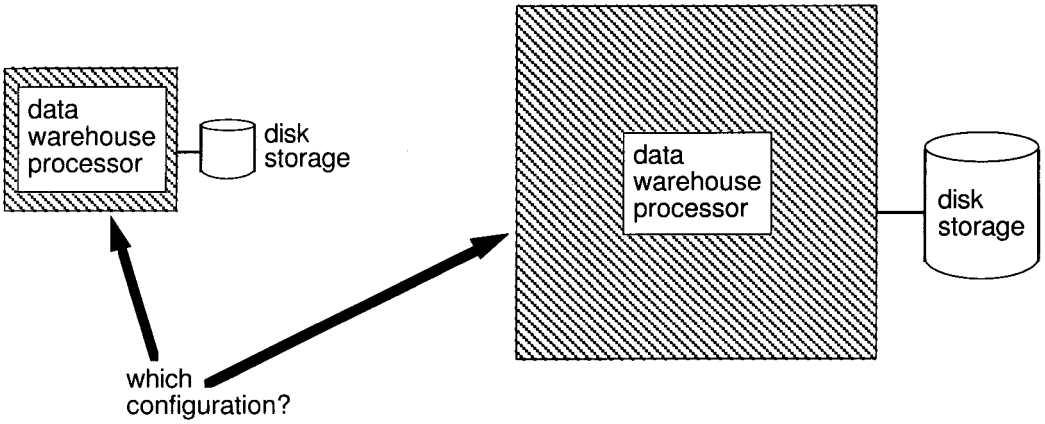


Figure 5: The two main aspects of capacity planning for the data warehouse environment are disk storage and processor size.

The question facing the capacity planner is - how much of these resources will be required on the one year and the five year horizon. As has been stated before, there is an indirect (yet very real) relationship between the volume of data and the processor required.

CALCULATING DISK STORAGE

The calculations for space are almost always done exclusively for the current detailed data in the data warehouse. (If you are not familiar with the different levels of data in the warehouse, please refer to the Tech Topic on the description of the data warehouse.) The reason why the other levels of data are not included in this analysis is that:

- they consume much less storage than the current detailed level of data, and
- they are much harder to identify.

Therefore, the considerations of capacity planning for disk storage center around the current detailed level.

The calculations for disk storage are very straightforward. Figure 6 shows the elements of calculation.

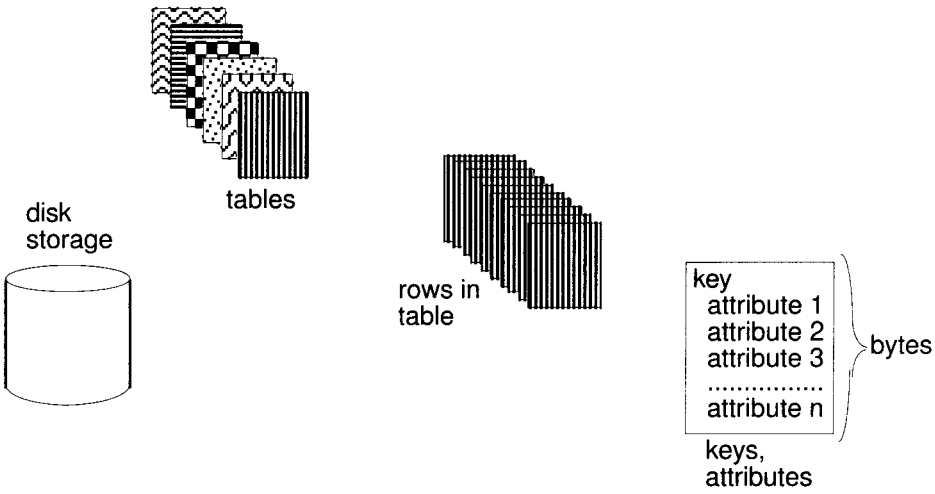


Figure 6: Estimating disk storage requirements for the data warehouse.

To calculate disk storage, first the tables that will be in the current detailed level of the data warehouse are identified. Admittedly, when looking at the data warehouse from the standpoint of planning, where little or no detailed design has been done, it is difficult to divine what the tables will be. In truth, only the very largest tables need be identified. Usually there are a finite number of those tables in even the most complex of environments.

Once the tables are identified, the next calculation is how many rows will there be in each table. Of course, the answer to this question depends directly on the granularity of data found in the data warehouse. The lower the level of detail, the more the number of rows.

In some cases the number of rows can be calculated quite accurately. Where there is a historical record to rely upon, this number is calculated. For example, where the data warehouse will contain the number of phone calls made by a phone company's customers and where the business is not changing dramatically, this calculation can be made. But in other cases it is not so easy to estimate the number of occurrences of data.

One approach is to look across the industry and see what other companies have experienced. This approach is quite effective if you can find out information that other companies are willing to share and if the company has a similar profile. Unfortunately, often times a comparative company is hard to find.

A third approach is to estimate the number of occurrences based on business plans, economic forecasts, and using the advice of specialized industry consultants. This is the least accurate method and the most expensive, but sometimes it is the only choice.

The number of occurrences of data is a function of more than just the business. The other factors are:

- the level of detail the data will be kept at, and
- the length of time the data will be kept.

After the number of rows are divined, the next step is to calculate the size of each row. This is done by estimating the contents of each row - the keys and the attributes.

Once the contents of the row are taken into consideration, the indexes that are needed are factored in.

As a matter of practice, very little if any free space is left in the data warehouse because data is not updated in the warehouse. In most circumstances, any free space in a data warehouse is wasted.

The total disk requirements then are calculated by adding all the requirements mentioned.

PROCESSOR REQUIREMENTS

In order to make sense of the estimation of the processor requirements for the data warehouse, the work passing through the data warehouse processor must be divided into one of three categories - background processing, predictable DSS processing, and unpredictable DSS processing.

Figure 7 shows these three categories.

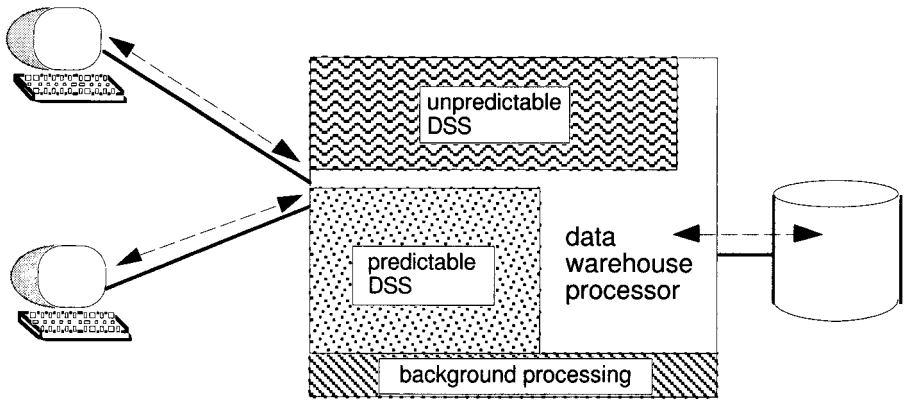


Figure 7: The three types of processing that occur in the data warehouse.

Background processing is that processing that is done on a predictable, (usually) batch basis. Typical of background processing is extract processing, data warehouse loads, monitors, sorts/merges, restructuring, index creations, etc. Background processing is that utilitarian processing necessary to the data warehouse but not directly associated with a query or an analysis of data warehouse data.

Background processing can be run at off peak times and can be spread evenly throughout the day. There is seldom much of a time constraint for background processing.

Predictable DSS processing is that processing that is regularly done, usually on a query or transaction basis. Predictable DSS processing may be modeled after DSS processing done today but not in the data warehouse environment. Or predictable DSS processing may be projected, as are other parts of the data warehouse workload.

The parameters of interest for the data warehouse designer (for both the background processing and the predictable DSS processing) are:

- the number of times the process will be run,
- the number of I/Os the process will use,
- whether there is an arrival peak to the processing,
- the expected response time.

These metrics can be arrived at by examining the pattern of calls made to the dbms and the interaction with data managed under the dbms.

The third category of process of interest to the data warehouse capacity planner is that of the unpredictable DSS analysis. The unpredictable process by its very nature is much less manageable than either background processing or predictable DSS processing. However, certain characteristics about the unpredictable process can be projected (even for the worst behaving process.) For the unpredictable processes, the:

- expected response time (in minutes, hours, or days) can be outlined,
- total amount of I/O can be predicted, and
- whether the system can be quiesced during the running of the request can be projected.

Once the workload of the data warehouse has been broken into these categories, the estimate of processor resources is prepared to continue.

The next decision to be made is whether the eight hour daily window will be the critical processing point or whether overnight processing will be the critical point. Usually the eight hour day - from 8:00 am to 5:00 pm, as the data warehouse is being used - is the critical point.

Assuming that the eight hour window is the critical point in the usage of the processor, a profile of the processing workload is created.

THE WORKLOAD MATRIX

The workload matrix is a matrix that is created as the intersection of the tables in the data warehouse and the processes (the background and the predictable DSS processes) that will run in the data warehouse.

Figure 8 shows a matrix formed by tables and processes.

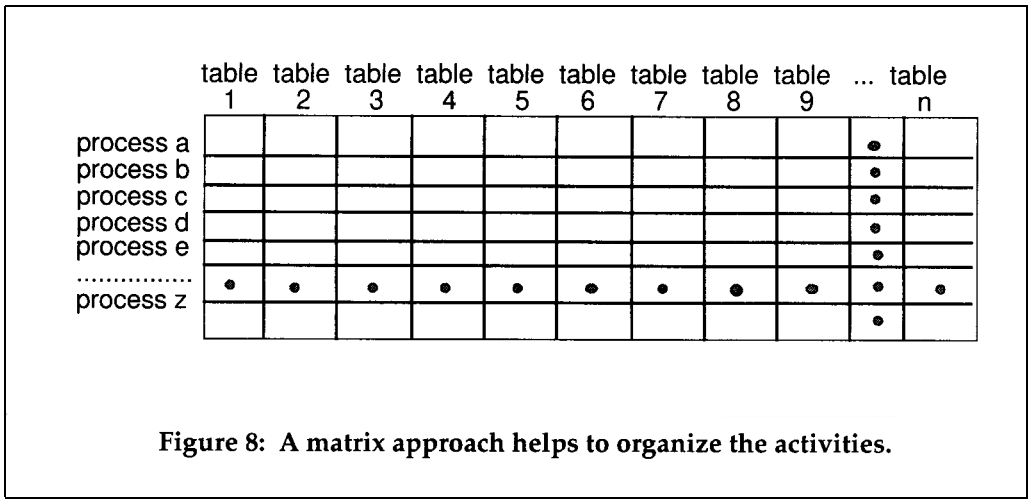


Figure 8: A matrix approach helps to organize the activities.

The workload matrix is then filled in. The first pass at filling in the matrix involves putting the number of calls and the resulting I/O from the calls the process would do if the process were executed exactly once during the eight hour window. Figure 9 shows a simple form of a matrix that has been filled in for the first step.

	table 1	table 2	table 3	table 4	table 5	table 6	table 7	table 8	table 9	...	table n
process a	2/5			2/10		5/15				•	3/9
process b	1/3		3/9						1/2	•	
process c		5/25					4/12		1/5	•	2/5
process d	1/6		4/8		1/2	3/9		3/9		•	1/2
process e	1/5		3/4		2/2	6/12		6/12		•	
.....	•	•	•	•	•	•	•	•	•	•	•
process z	1/5	5/25	4/8		1/2				5/15	•	5/15

number of calls/i/os per access

Figure 9: The simplest form of a matrix is to profile individual transactions in terms of calls and I/Os.

For example, in Figure 9 the first cell in the matrix - the cell for process a and table 1 - contains a "2/5". The 2/5 indicates that upon execution process a has two calls to the

table and uses a total of 5 I/Os for the calls. The next cell - the cell for process a and table 2 - indicates that process a does not access table 2.

The matrix is filled in for the processing profile of the workload as if each transaction were executed once and only once.

Determining the number of I/Os per call can be a difficult exercise. Whether an I/O will be done depends on many factors:

- the number of rows in a block,
- whether a block is in memory at the moment it is requested,
- the amount of buffers there are,
- the traffic through the buffers,
- the dbms managing the buffers,
- the indexing for the data,
- the other part of the workload,
- the system parameters governing the workload, etc.

There are, in short, MANY factors affecting how many physical I/O will be used.

The I/Os an be calculated manually or automatically (by software specializing in this task.)

After the single execution profile of the workload is identified, the next step is to create the actual workload profile. The workload profile is easy to create. Each row in the matrix is multiplied by the number of times it will execute in a day. The calculation here is a simple one.

Figure 10 shows an example of the total I/Os used in an eight-hour day being calculated.

	table 1	table 2	table 3	table 4	table 5	table 6	table 7	table 8	table 9	...	table n
process a	2500			5000		7500					4500
process b	1000		3000						666		
process c		2500					1200		500		500
process d	750		1000		250	1250		1250			250
process e	700		566		283			1750			
.....											
process z	750	3750	1000		300				2250		2250
total I/O	15000	85000	12250	10250	5750	27750	19500	6750	8926		87560

total I/Os

Figure 10: After the individual transactions are profiled, the profile is multiplied by the number of expected executions per day.

At the bottom of the matrix the totals are calculated, to reach a total eight-hour I/O requirement.

After the eight hour I/O requirement is calculated, the next step is to determine what the hour-by-hour requirements are. If there is no hourly requirement, then it is assumed that the queries will be arriving in the data warehouse in a flat arrival rate. However, there usually are differences in the arrival rates. Figure 11 shows the arrival rates adjusted for the different hours in the day.

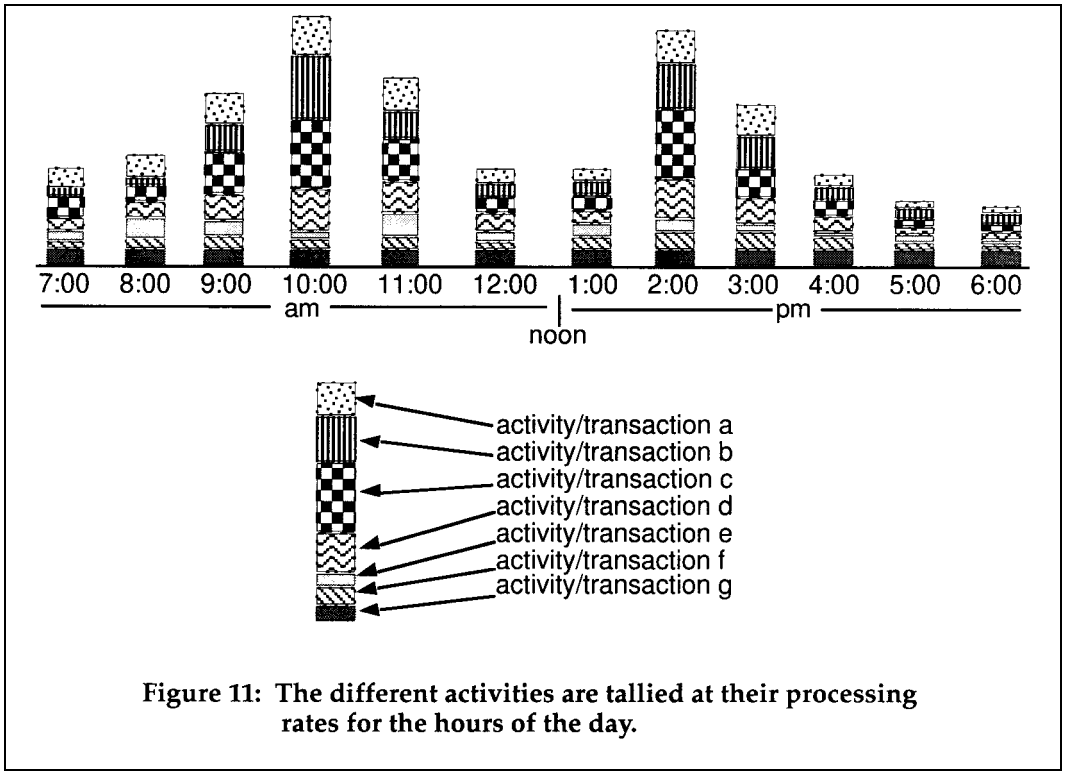


Figure 11: The different activities are tallied at their processing rates for the hours of the day.

Figure 11 shows that the processing required for the different identified queries is calculated on an hourly basis.

After the hourly calculations are done, the next step is to identify the "high water mark." The high water mark is that hour of the day when the most demands will be made of the machine. Figure 12 shows the simple identification of the high water mark.

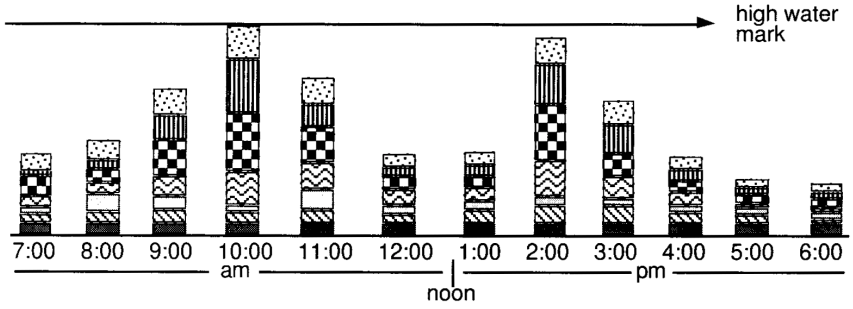


Figure 12: The "high water mark" for the day is determined.

After the high water mark requirements are identified, the next requirement is to scope out the requirements for the largest unpredictable request. The largest unpredictable request must be parameterized by:

- how many total I/Os will be required,
- the expected response time, and
- whether other processing may or may not be quiesced.

Figure 13 shows the specification of the largest unpredictable request.

next the largest unpredictable activity is sized:
- in terms of length of time for execution
- in terms of total I/O

Figure 13

After the largest unpredictable request is identified, it is merged with the high water mark. If no quiescing is allowed, then the largest unpredictable request is simply added as another request. If some of the workload (for instance, the predictable DSS processing) can be quiesced, then the largest unpredictable request is added to the portion of the workload that cannot be quiesced. If all of the workload can be quiesced, then the unpredictable largest request is not added to anything.

The analyst then selects the larger of the two - the unpredictable largest request with quiescing (if quiescing is allowed), the unpredictable largest request added to the portion of the workload that cannot be quiesced, or the workload with no unpredictable processing. The maximum of these numbers then becomes the high water mark for all DSS processing.

Figure 14 shows the combinations.

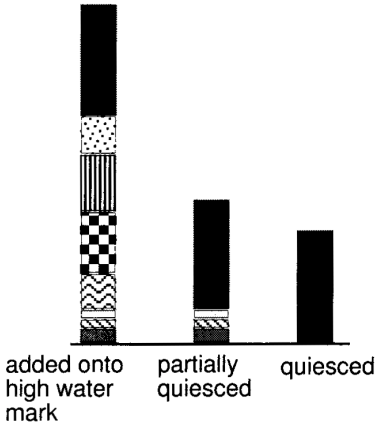


Figure 14: The impact of the largest unpredictable request is estimated.

The maximum number then is compared to a chart of mips required to support the level of processing identified, as shown in Figure 15.

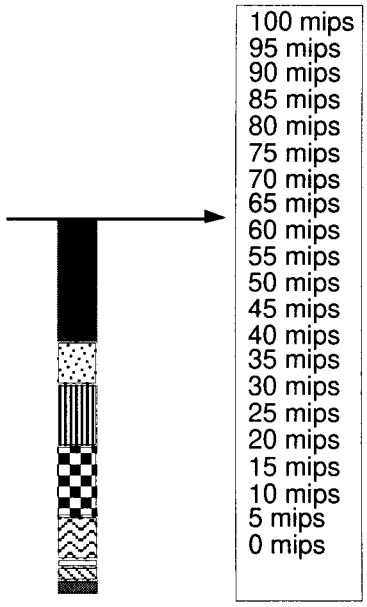


Figure 15: Matching the high water mark for all processing against the required MIPS.

Figure 15 merely shows that the processing rate identified from the workload is matched against a machine power chart.

Of course there is no slack processing factored. Many shops factor in at least ten percent. However, factoring an unused percentage may satisfy the user with better response time, but costs money in any case.

The analysis described here is a general plan for the planning of the capacity needs of a data warehouse. It must be pointed out that the planning is usually done on an iterative basis. In other words, after the first planning effort is done, another more refined version soon follows.

In all cases it must be recognized that the capacity planning effort is an estimate.

SUMMARY

Capacity planning is important for the data warehouse environment as it was (and still is!) for the operational environment. Capacity planning in the data warehouse environment centers around planning disk storage and processing resources.

There is an important but indirect relationship between data and processing power - the more data there is, the more the processing power required.

Processing in the data warehouse environment must be physically separated from processing in the operational environment.

Disk storage capacity is a function of the level of detail stored, the length of time the data is kept, and the number of occurrences of data to be stored.

Processor capacity is a function of the workload passing through the environment. The important features of the processing environment are the characteristics of the workload, which can be described as consisting of background processing, predictable DSS processing, and unpredictable DSS processing. The profile of the transactions are merged together to produce a definitive picture of the resources needed for the data warehouse environment.