

THE DATA WAREHOUSE BUDGET

BY

W. H. INMON

“Anyone who lives within his means suffers from a lack of imagination”
Lineal Standee, 1967

“Solvency is entirely a matter of temperament and not of income”
Logan Pearsall Smith, 1931

Information systems is experiencing a major paradigm shift the magnitude of which has not been seen since the early days of database in the 1960s. The paradigm shift is to an architecture centered on a data warehouse. The data warehouse has captured the imagination and attention of the information systems community for good reason. It appeals in equal measure to the maintenance programmer struggling with old legacy systems and to the businessperson needing to use information for better decisions. The data warehouse is unique in that it intersects the interests of both the technician and the businessperson in a manner and to a degree that has never before been experienced.

DATA WAREHOUSE COSTS

Like all other structures of information that preceded it, the data warehouse has its own peculiarities and costs. This Tech Topic will explore what the data warehouse costs are, and will develop a model for anticipating and managing those costs.

THE DATA WAREHOUSE ENVIRONMENT

In order to understand the costs associated with data warehouse, Figure 1 shows a simple model that outlines the different components of the warehouse environment.

There are several terms used to describe the data warehouse environment; some of these terms may be unfamiliar. The first is “terminal analysis.” Terminal analysis as used in this Tech Topic refers to the work done to analyze and display data once the data is in the data warehouse. Terminal analysis has many components:

- discover,
- query preparation,
- query management,
- query interpretation,
- result set analysis,
- result display, and so forth.

Terminal analysis then is the analytical activity that occurs after the data is placed in the data warehouse. Terminal analysis occurs at many levels - the executive level, the mid-level analytical level, the detailed level, and so forth.

Ongoing refreshment of data in the data warehouse refers to the activity of selecting and processing legacy data for periodic refreshment into the data warehouse. There may be many approaches to ongoing refreshment. The most common approach is the access and manipulation of the log or journal tapes, which are produced during operational processing. It is very efficient to be able to read a log tape off line and determine where changes need to be made in the data warehouse.

THE DATA WAREHOUSE BUDGET

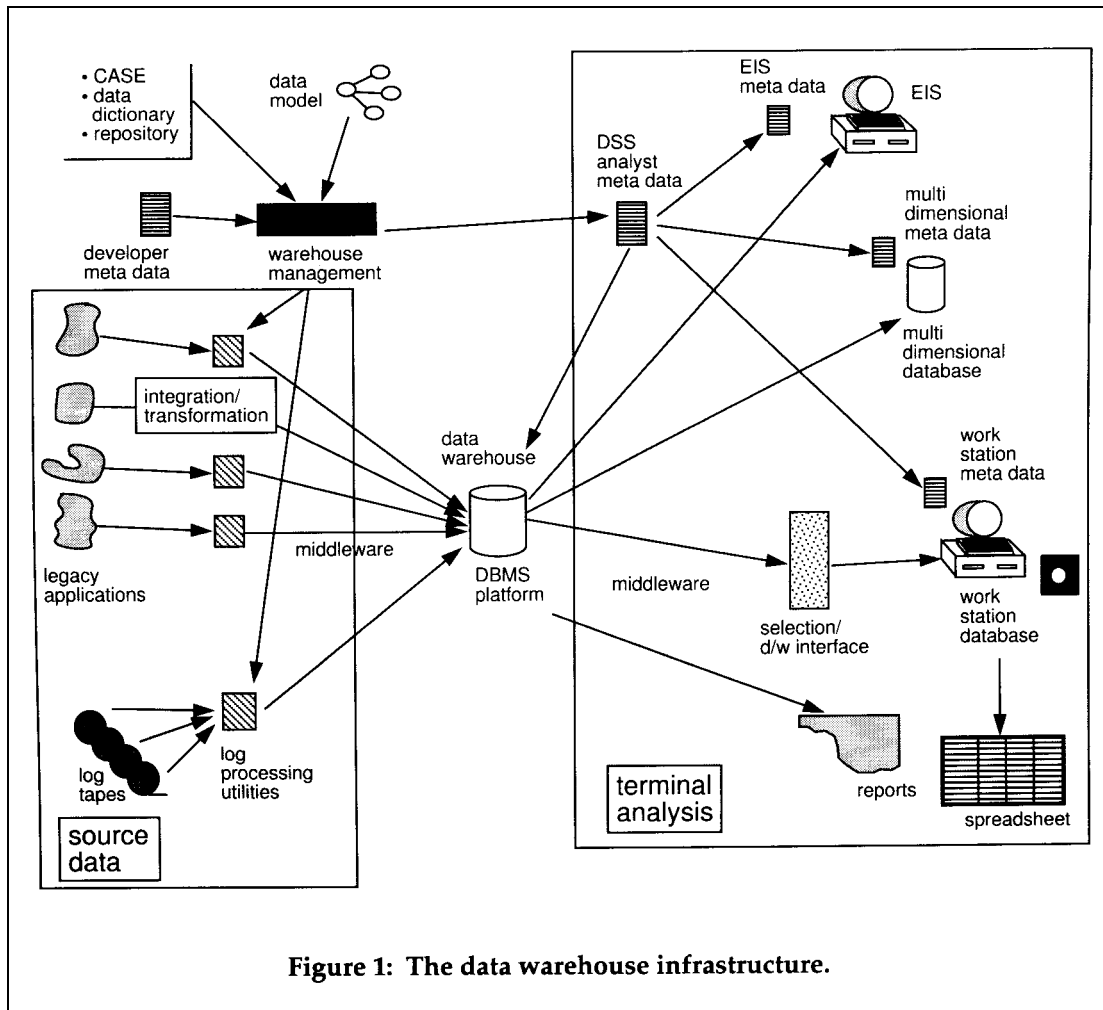


Figure 1: The data warehouse infrastructure.

Figure 1: The data warehouse infrastructure.

Integration and transformation refers to the programs, which need to be written in order to properly move data from the legacy operational environment to the data warehouse. At first glance these programs appear to be simple extract programs. They are not. There is a large and complex amount of work that needs to be done to make the transformation from one environment to the next. The transformation programs include functionality that:

- filters data,
- summarizes data,
- reformats data,
- changes DBMS technology,
- selects data from multiple sources,
- changes the keys of data,
- resequences data, and so forth.

The design and analytical effort begins with a data model. The data model serves as the roadmap for integration of the different aspects of the operational world. The data model is usually housed on some form of CASE or repository technology.

THE DATA WAREHOUSE BUDGET

There is a meta data infrastructure that exists for the developer and for the decision support systems (DSS) analyst. In addition there may or may not be a development suite of tools specifically for the creation and maintenance of the data warehouse environment.

The system of record is the definition of the legacy source data. Data is moved to the data warehouse and is moved out of the data warehouse by means of middleware.

The data warehouse itself can reside on a variety of platforms - mainframes, parallel platforms or client/server platforms. And there are a wide variety of DBMS that can be used to house and manage the data warehouse data itself.

CLASSIFYING THE COSTS

The costs for the data warehouse environment begin with the breaking of the costs into two categories - recurring or ongoing costs and one-time initial costs. The one-time costs can be further categorized into hardware and software costs. In addition the costs can be categorized as capital costs and operational costs. A capital cost is one that is for the purchase of a specific asset. An operational cost is one that is expended for manpower in the building and usage of the data warehouse environment.

The different categories of expenditures can be grouped into a matrix as seen in Figure 2.

| | | RECURRING | ONE TIME | | |
|---|--|---|---|---|--|
| CAPITAL | | <ul style="list-style-type: none"> • hardware maintenance • software maintenance • terminal analysis • middleware | <table style="width: 100%; border: none;"> <tr> <td style="vertical-align: top;"> <ul style="list-style-type: none"> hardware • disk • CPU • network • terminal analysis </td> <td style="vertical-align: top;"> <ul style="list-style-type: none"> software • DBMS • terminal analysis • middleware • network • log utility processing • meta data infrastructure </td> </tr> </table> | <ul style="list-style-type: none"> hardware • disk • CPU • network • terminal analysis | <ul style="list-style-type: none"> software • DBMS • terminal analysis • middleware • network • log utility processing • meta data infrastructure |
| <ul style="list-style-type: none"> hardware • disk • CPU • network • terminal analysis | <ul style="list-style-type: none"> software • DBMS • terminal analysis • middleware • network • log utility processing • meta data infrastructure | | | | |
| OPERATIONAL | | <ul style="list-style-type: none"> • ongoing refreshment • integration transformation maintenance • data model maintenance • system of record identification maintenance • meta data infrastructure maintenance • archival of data • aging of data within data warehouse | <ul style="list-style-type: none"> • integration/transformation processing specification • meta data infrastructure population • system of record definition • DDL definition • network transfer definition • CASE/repository interface • initial data warehouse population • data model/database design and definition | | |

Figure 2: The different kinds of expenditures that are made for the building and usage of the data warehouse.

Figure 2: The different kind of expenditures that are made for the building and usage of the data warehouse.

ONE-TIME EXPENSES

The analysis of expenses is best begun with the consideration of the initially occurring, one-time expenses. The data warehouse is built and maintained in an iterative manner. Each iteration of data warehouse development proceeds down a path called "subject area" development. Subject areas center around things such as CUSTOMERS, PRODUCTS, SALES, VENDORS, etc. First one subject area is developed, then another is developed, and so forth. The track record of data warehouse development success strongly suggests that a data warehouse NOT be built in a massive application development style, where a large amount of subjects are developed at once.

The first iteration of warehouse development starts with the development and population of a single subject area. As the first subject area is built, one-time expenses are incurred. The next iteration of development focuses on another subject area, and so forth. The need to build the data warehouse iteratively is reflected by the pattern of expenditures for data warehouse development. Consider the pattern of development expenditure as shown in Figure 3.

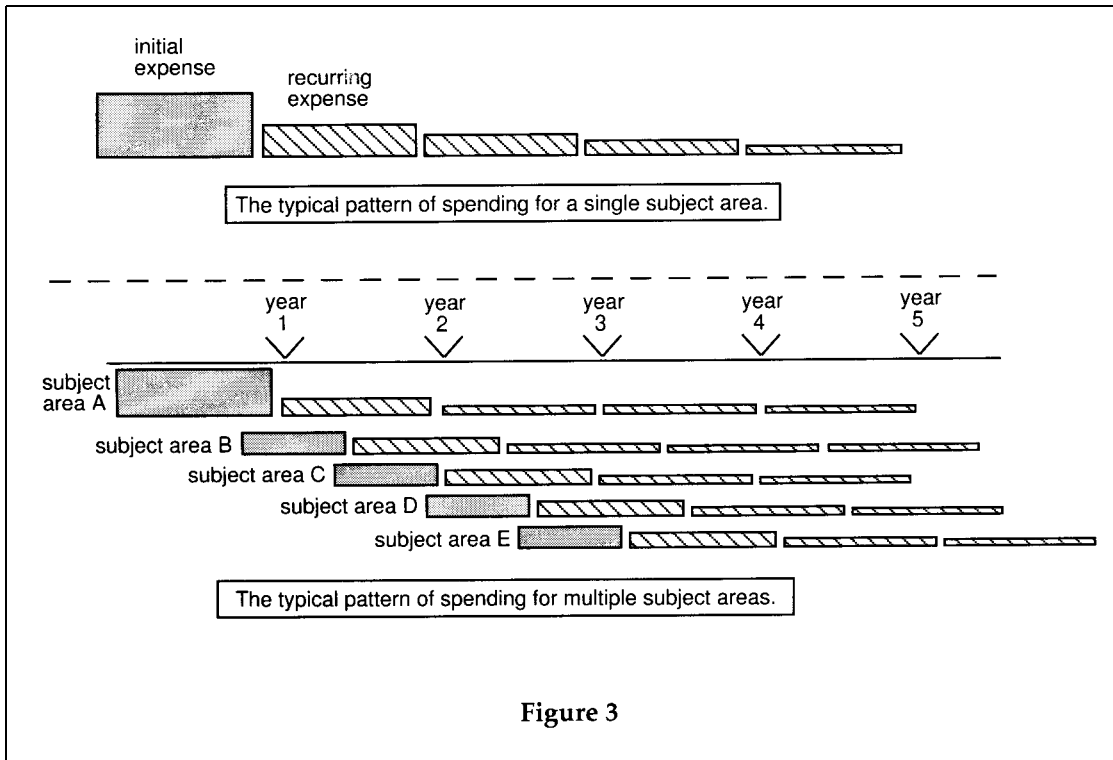


Figure 3

Figure 3 shows that initially, development and population expenses are high. As time passes recurring expenses drop for the building and maintenance of the data warehouse. From year to year the recurring expenses of modification and population of the data warehouse recede for a given subject area.

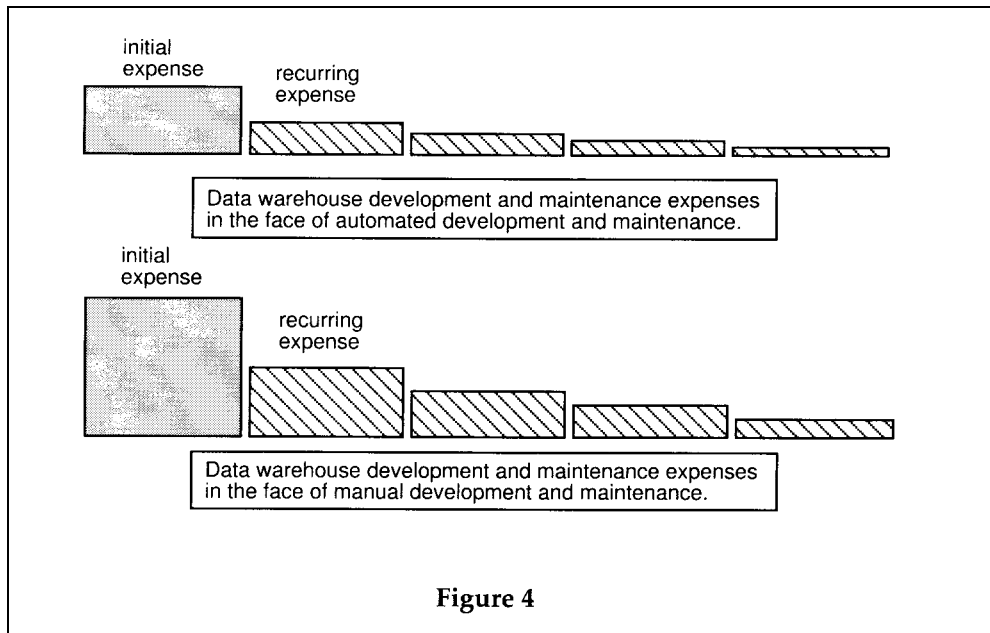
The bottom half of Figure 3 shows that multiple subject areas are built and maintained over time. The initial start up expense is not as large, as there is some economy of scale to be enjoyed. Once the hardware platform is broken in, once the DBMS is built and

THE DATA WAREHOUSE BUDGET

support procedures are put into place, and once the user is trained, there is no need to repeat these activities.

The time sequencing of both the one-time and recurring expenditures is a function of the ambition of the data architect. Based on the ambition of the data warehouse development schedule and the degree of economy of scale that can be enjoyed, it is seen that over time there is a decreasing cost associated with data warehouse development and maintenance.

There are, however, some factors affecting the size of the expenditures for data warehouse development. The most profound decision is whether to build the data warehouse manually or automatically. Tools for automating data warehouse development have been built because of the extreme similarity of work required for accessing and converting the legacy systems environment into the data warehouse. While this process must be done properly, it is complex, tedious and rote work that varies little from one company to the next. The same work that is done over and over can be automated, and in doing so, MASSIVE amounts of labor can be saved. The savings of labor shows up in BOTH the development costs and in the maintenance costs. Figure 4 shows that initial and recurring expenses are profoundly affected by the decision to build the data warehouse manually or in an automated fashion.



The initial and recurring expenses for the building and maintenance of the data warehouse are significantly raised by choosing to build the data warehouse manually. In addition, the length of time required to build the data warehouse is significantly extended by choosing to build the warehouse manually.

Another factor significantly affecting the expenditures made on behalf of the data warehouse is that of the platform chosen for the warehouse. The two popular choices are a parallel platform and a client/server platform. The parallel choice is usually made

THE DATA WAREHOUSE BUDGET

because of the need to manage a very large amount of data. The client/server choice is made when there is less than a very large amount of data. From the standpoint of the hardware platform cost, client/server costs are less than parallel costs. For the purpose of comparing different patterns of expenditures, four models will be developed:

- a parallel processing model where automated development is done,
- a client/server model where automated development is done,
- a parallel processing model where manual development is done, and
- a client/server model where manual development is done.

Figures 5, 6, 7 and 8, show these different models.

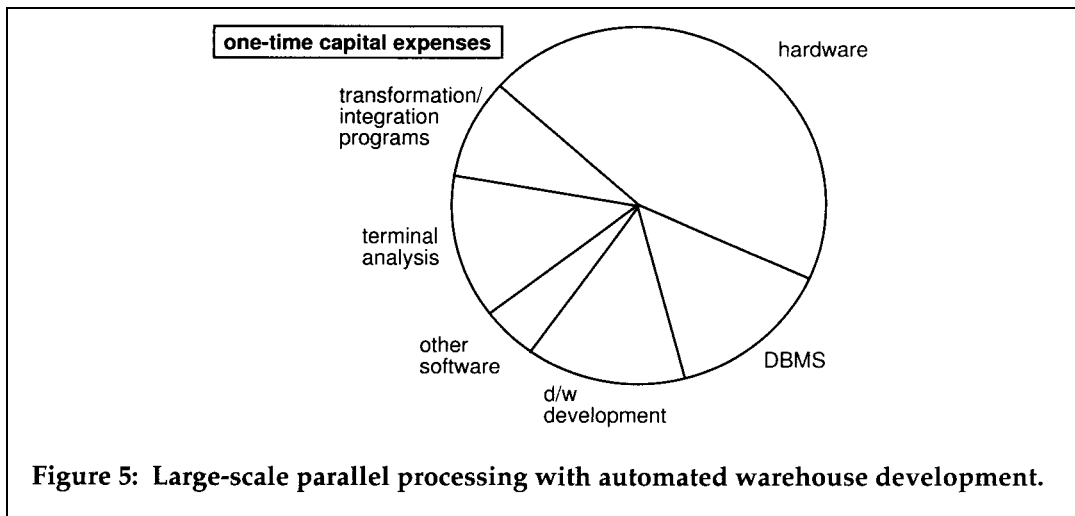


Figure 5: Large-scale parallel processing with automated warehouse development.

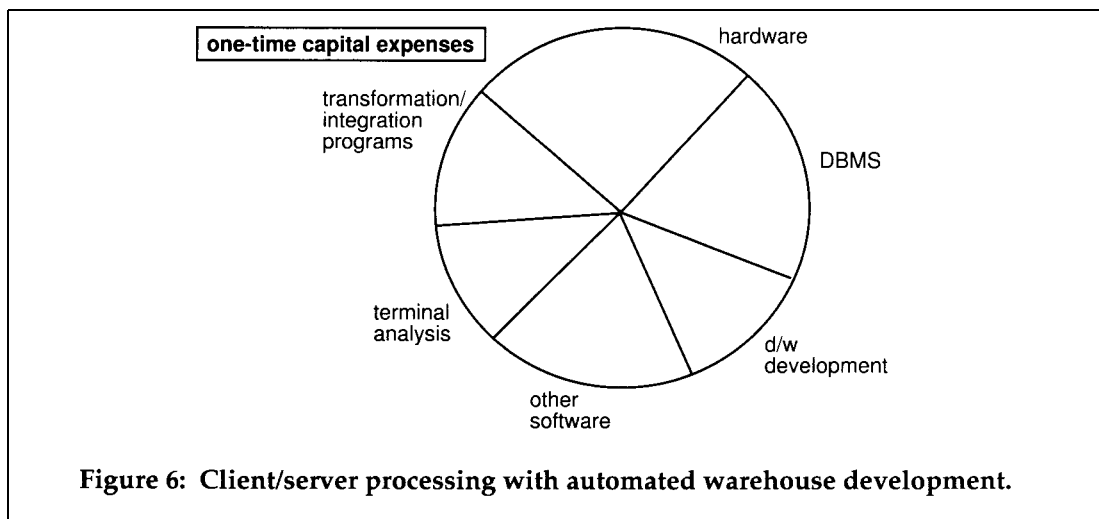


Figure 6: Client/server processing with automated warehouse development.

THE DATA WAREHOUSE BUDGET

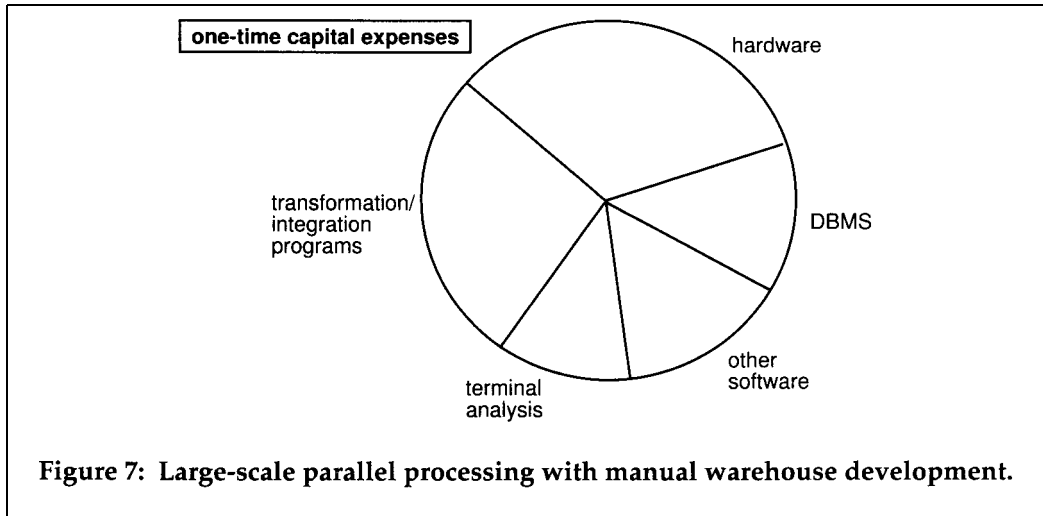


Figure 7: Large-scale parallel processing with manual warehouse development.

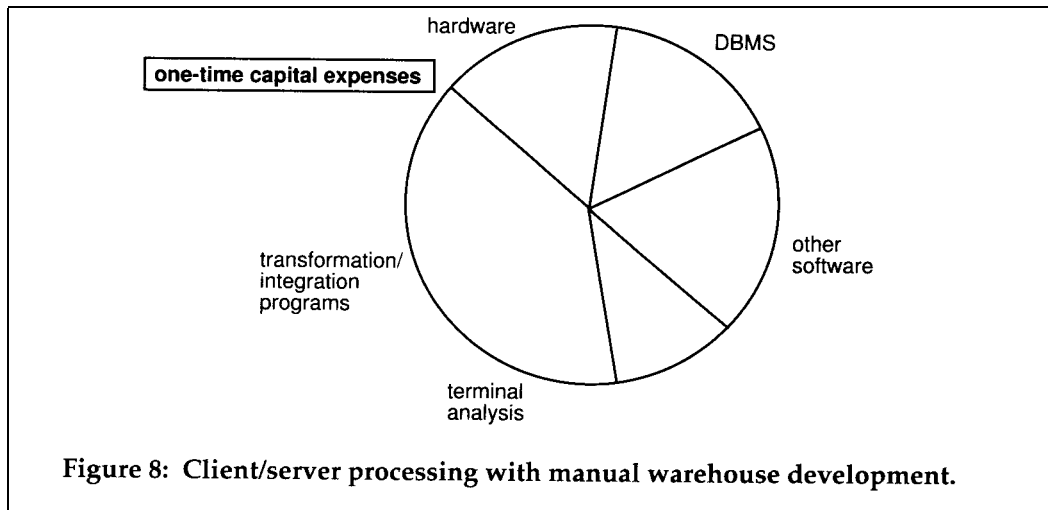


Figure 8: Client/server processing with manual warehouse development.

The parallel automated model shows that hardware is easily the most expensive part of the model for one-time capital expenditures. DBMS and other software costs are significant, but the lion's share of the budget is for hardware platforms. Because the environment is automated, there is a cost for the data warehouse development tool. The client/server automated environment is one where hardware consumes a smaller part of the budget. All other costs go up proportionately because there is less money required in total for this environment.

The parallel manual environment shows that hardware consumes a major part of the budget. There are of course no cost for data warehouse development tools. But the integration and transformation programs that have to be written and maintained manually start to consume a significant part of the budget.

THE DATA WAREHOUSE BUDGET

In the case of client/server manual data warehouse development, hardware consumes even less of the budget, and integration and transformation programs consume even more of the budget.

The models that are shown in Figures 5, 6,7 and 8 show the relative expenditures within the different models that have been developed. But how the models themselves compare to each other is another matter entirely. Figure 9 shows how the different models compare in terms of total budget.

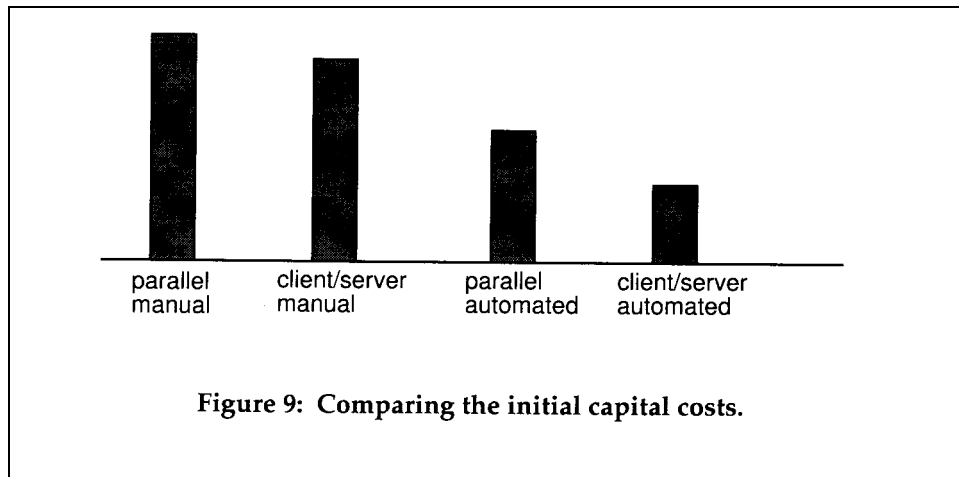


Figure 9: Comparing the initial capital costs.

Figure 9 Shows that the parallel manual approach is the most expensive. The client/server automated approach is the least expensive. The client/server manual approach is less expensive than the parallel manual approach. And the parallel automated approach is less than the client/server manual approach.

One conclusion that can be drawn from Figure 9 is that the client /server approach is less expensive than the parallel approach. From a hardware perspective that is true. However, when the size of the data in the warehouse is considerable, there may not be a choice as to the platform that can be chosen.

RECURRING EXPENSES

The primary expenditures for recurring expenses are hardware, terminal analysis and refreshment. Hardware expenses include both incremental new acquisitions and maintenance on existing hardware. Terminal analysis is a significant expense, but the larger the expense here, the more successful the data warehouse. Refreshment expense deserves an explanation. The data warehouse needs to be periodically refreshed with new data as the operational systems are updated. Sometimes this update is made as often as every 24 hours. In other cases this update is made weekly or monthly. One way to accomplish this update is go to the native operational DBMS and pull the data from there. In most cases going back to the native DBMS causes many records to be read that have had no activity occur against them. In this case the same data is read repeatedly, even though most of the data has had no changes made to it. In any case,

going back to the native operational DBMS causes the mainline operational system to be up, running, and online. Such an approach to the refreshment of data warehouse data is wasteful and expensive.

A much more efficient approach is to read the log or journal tapes that have been created during the operational online transaction processing. The log tape contains only the data that has changed. And the log tape can be processed off line, away from the operational online system. Doing refreshment from log tapes is a MUCH more efficient way to accomplish refreshment. However, log tapes are notoriously difficult to read. Special utilities are required to read data from the log tape and prepare it for entry into the data warehouse.

The second major issue facing the data architect in the management of recurring expenses is that of whether the integration and transformation programs have been developed automatically or in a manual manner. If the integration and transformation programs have been developed manually, they will have to be maintained manually. On the other hand, if the integration and transformation programs have been developed automatically, they can be maintained automatically. The integration and transformation programs in the data warehouse are notorious for needing maintenance. A maintenance change is needed every time the operation environment changes, every time the transformation logic changes, and every time the data warehouse itself changes. In other words, the integration and transformation interface between the data warehouse and the legacy systems environment is very unstable. The instability means that with the operation of the data warehouse - in a mature state - recurring expenses of maintenance can be large if done manually.

Another issue relating to an automated integration/transformation interface is that of the creation of the meta data infrastructure. For users to be able to access and use the data in the data warehouse effectively, a data warehouse directory is needed. The data warehouse directory is similar in concept to the card catalog at a public library except that it applies to the data warehouse. When the integration and transformation are done automatically, the meta data infrastructure is also created and maintained automatically. When the integration and transformation programs are written manually, the meta data infrastructure must be built and maintained manually as well.

Other ongoing expenses include the amount of money required for DBMS maintenance.

Figures 10, 11, 12 and 13 show the different expense models that relate to the styles of performing ongoing maintenance for the data warehouse.

THE DATA WAREHOUSE BUDGET

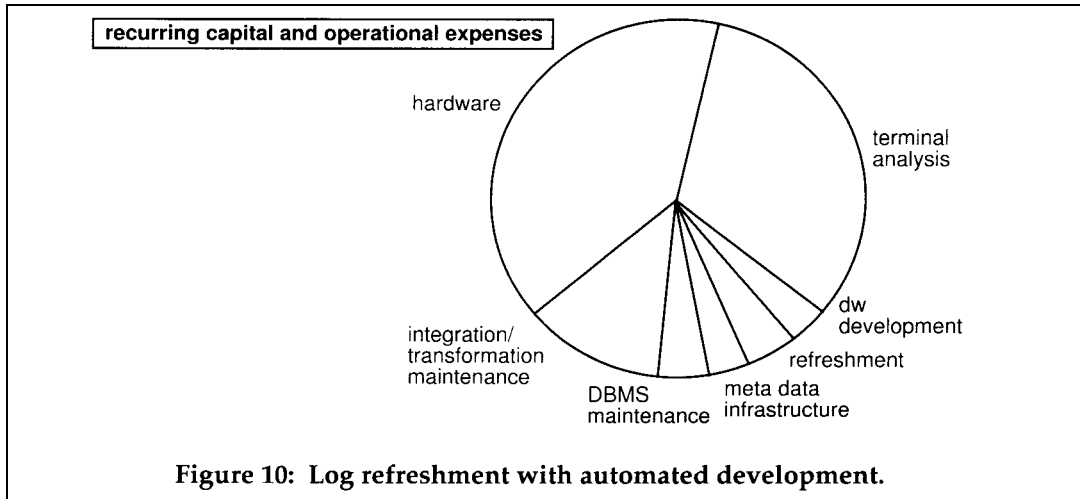


Figure 10: Log refreshment with automated development.

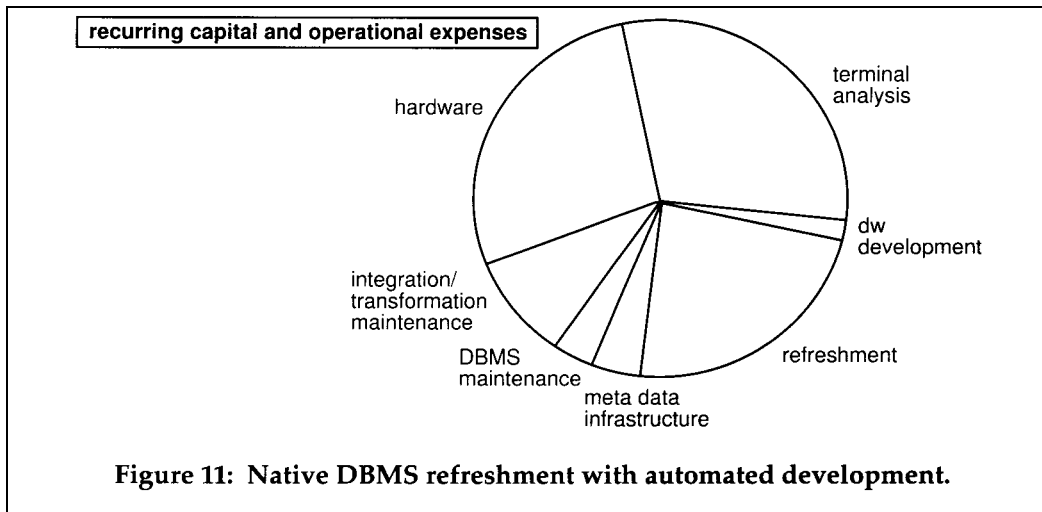


Figure 11: Native DBMS refreshment with automated development.

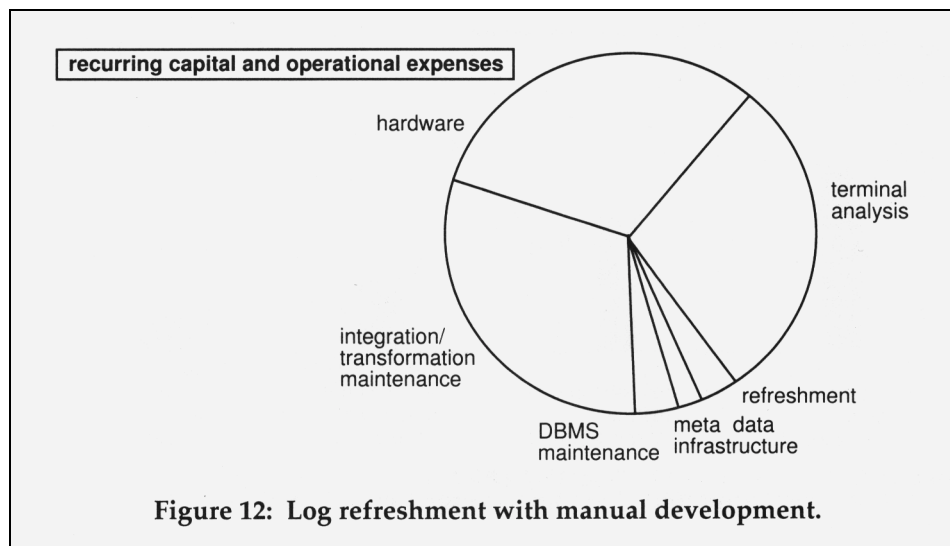


Figure 12: Log refreshment with manual development.

Figure 12: Log refreshment with manual development.

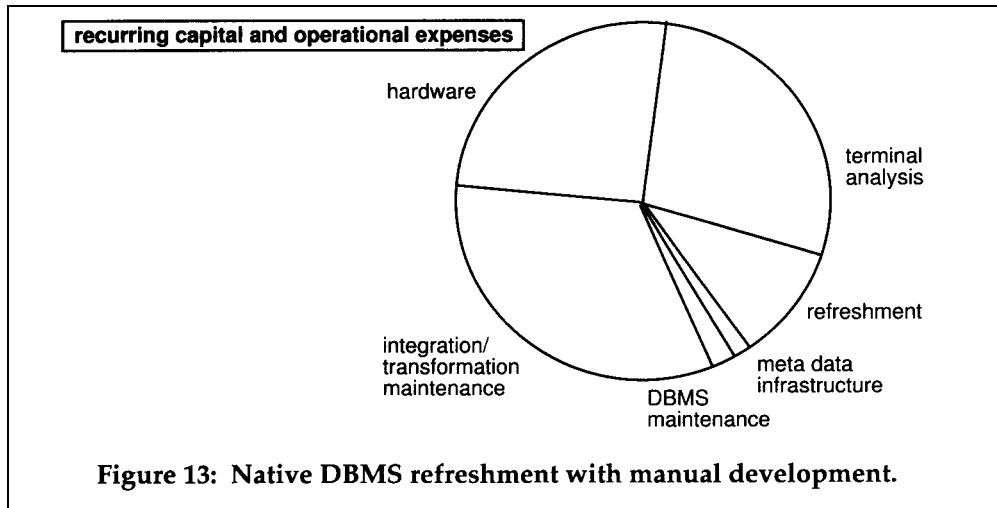


Figure 13: Native DBMS refreshment with manual development.

The first model shows the relative expenses when log refreshment is done, and development and maintenance are performed in an automated manner. Hardware and terminal analysis take up the largest percentages of the budget.

When native DBMS refreshment is done along with automated development, hardware terminal analysis and refreshment consume most of the data warehouse budget. Note that refreshment becomes a very significant part of the budget when it is done for native DBMS access of the operational environment.

The next case is that of log refreshment in the face of manual development. In this case there is no cost of a data warehouse development environment, but the cost of maintaining the integration and transformation programs starts to sky rocket. Many more resources are needed for manual program maintenance than for automated maintenance.

The last case shows the relative expenditures when native DBMS refreshment is done with manual development and maintenance. Refreshment costs rise, as well as integration and transformation program maintenance.

THE DIFFERENT RECURRING COST MODELS

The relative costs of the different components are shown by the previous figures. However, what is not apparent is the total expenditure associated with each model. Figure 14 shows the total costs relative to each recurring expenditure model.

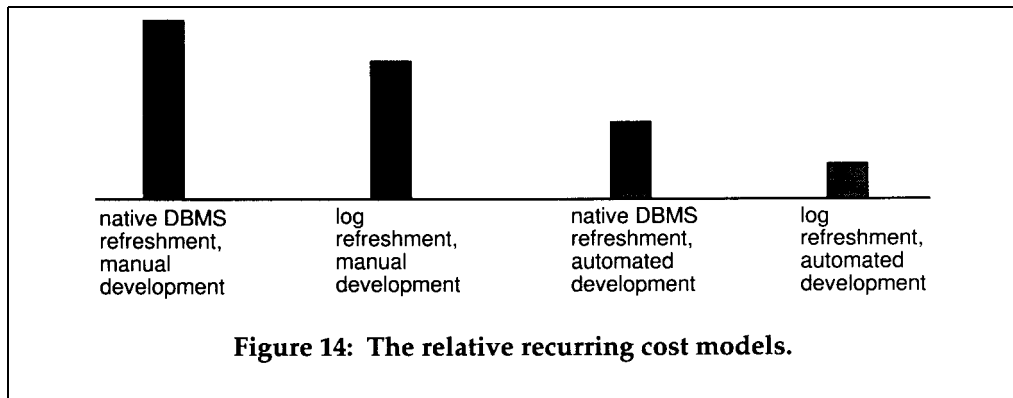


Figure 14: The relative recurring cost models.

Far and away the most expenditures occur for the case of native DBMS refreshment coupled with manual development. Going to log refreshment can lessen costs. Going to automated development and maintenance can significantly lower costs. By almost an order of magnitude, ongoing costs for data warehouse maintenance can be reduced by doing log tape refreshment and automated maintenance of integration and transformation programs.

CONTROLLING THE BUDGET

Easily the most two important things the manager can do to control the data warehouse budget are:

- build the data warehouse in iterative steps rather than in large development efforts (sometimes called “big bang” efforts), and
- carefully design the levels of granularity so that the most effective processing can be done at the highest levels of summarization.

If nothing else is done, these two approaches to data warehouse design and development will go a long way toward keeping the data warehouse budget under control. But there are other approaches to managing the budget.

CHARGEBACK

One of the most important disciplines that can be put into place in the controlling of the data warehouse budget is that of instigating a chargeback system. A chargeback system is useful in sending the message to the user community that resources are being consumed in the running and usage of the data warehouse environment. Such a message is a maturing influence even if the money being charged the user is only “funny money” that appears on a report.

Chargeback in the data warehouse is done on a basis of public and private charges. Public charges for the data warehouse are allocated for those expenses that everyone consumes. Private charges are those that are directly attributable to the processing done by a given user. Figure 15 outlines the difference between public and private charges in a data warehouse environment.

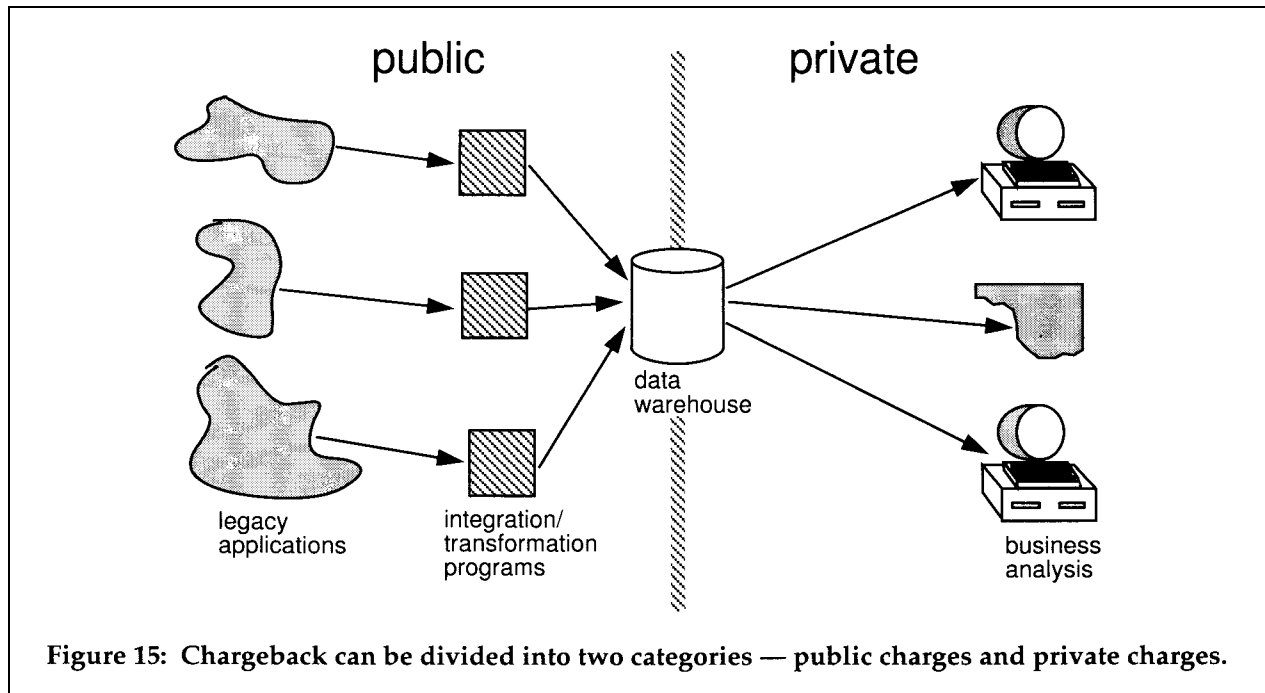


Figure 15: Chargeback can be divided into two categories - public charges and private charges.

Figure 15 shows that public expenses include such things as:

- the cost of disk storage,
- the cost of writing and maintaining integration and transformation programs,
- the cost of executing integration and transformation programs,
- the processor cost for the current detailed level of the data warehouse,
- the building and maintenance of the meta data infrastructure, and so forth.

The public expenses of the data warehouse environment include:

- the cost of executing a given query or accessing the information in the data warehouse.

Private data warehouse expenses are based on data trapped in system logs and monitors. The trapped data is based on the submission and execution of queries generated by the user. Private data warehouse expenses have many factors that determine just how much the cost of processing will be, such as:

- the amount of data requested,
- the priority of the request,
- the time of day the request was made,
- the time of day the response to the request is required,
- time of the month/week the request is made, and so forth.

There are advantages and disadvantages to using a chargeback system in the data warehouse environment. The disadvantage of a chargeback system is that it can discourage the user from ever experiencing the data warehouse. If the costs of

chargeback are too high, then the user may never discover what can be done with a data warehouse. For this reason chargeback needs to be done carefully and gently. On the other hand, with a charge back system the user learns that there is a cost associated with the business analysis being done. A corporation is placed in a precarious position when the user does not understand that there is a cost associated with informational processing. The data warehouse is a particularly poor place to not instill an attitude of conservation of resources because of its size. The volume of data that exists inside the data warehouse invites the user to experiment. This experimentation can lead to dramatic and very useful discoveries. However, there can be a high cost to the experimentation as well.

ELEVATING THE PROCESSING IN THE WAREHOUSE

When there is a sense of conservation of resources in a data warehouse environment, the user ultimately does as much processing as possible at as high a level of summarization as is available. Figure 16 shows the difference between having a chargeback system and not having a chargeback system in terms of the level of detail used by the user.

When there is no chargeback system the user attempts to operate as much as possible on as low a level of detail as possible. This is an extremely expensive attitude in the long run. As volumes of data mount, as the numbers of users grow who want to access the data, and as the requests for analysis grow in size, the technology that houses the data warehouse becomes stretched. The budget for hardware starts to climb exponentially at this point. But when the user is conditioned to do as much processing as possible at the highest level of summarization as possible, the technology that houses the data warehouse is used to its fullest and to its best advantage.

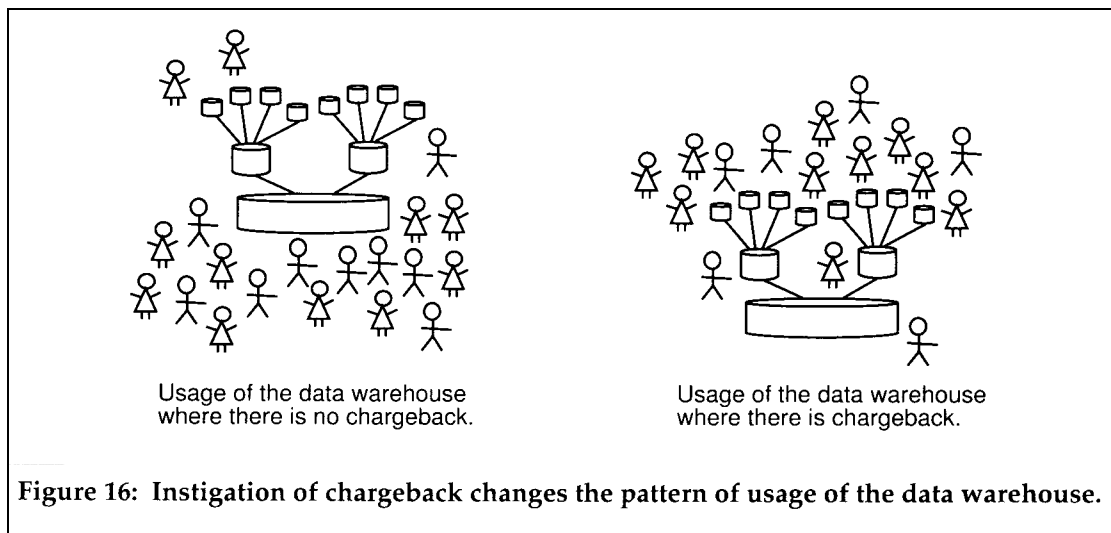


Figure 16: Instigation of chargeback changes the pattern of usage of the data warehouse.

DATA MARTS

Extending the notion that the user should be responsible for his/her own data warehouse budget is the idea of a data mart. A data mart is an extension of the data warehouse and is sometimes called a departmental machine. Multidimensional databases are often used in conjunction with data marts. With a data mart the user takes over direct control of a subset of the data warehouse. The subset of data is customized for the user as it enters the data mart. As the user builds and starts to use the data mart, the budget for the data mart shifts directly to the user. Figure 17 shows the movement of data and processing to the data mart.

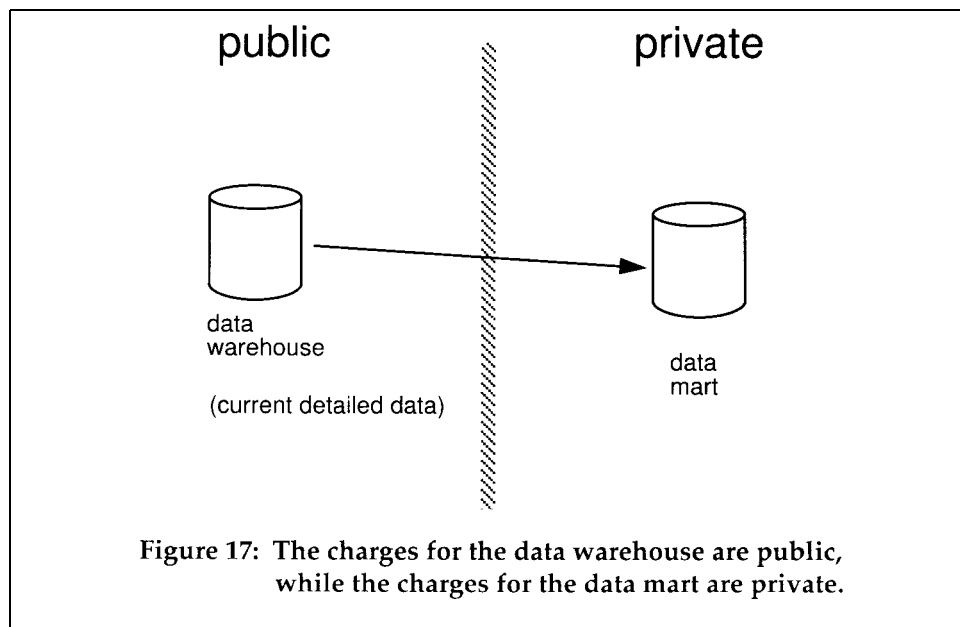


Figure 17: The charges for the data warehouse are public, while the charges for the data mart are private.

Data is typically summarized or otherwise customized to fit the user's specifications in the data mart.

CONSOLIDATED DATA ACCESS

One of the big expenditures for the data warehouse comes in the access of detailed data in the warehouse itself. This expenditure grows as the data warehouse becomes fully populated and as the number of users grows. There is an easy way to minimize this expenditure and that is to create a consolidated access of data. Figure 18 shows two scenarios. In one scenario everyone is doing his or her own individual access of data. The individual access is expensive. In the other scenario, the users have a common access mechanism that allows a single access of the data warehouse that can be shared among the users.

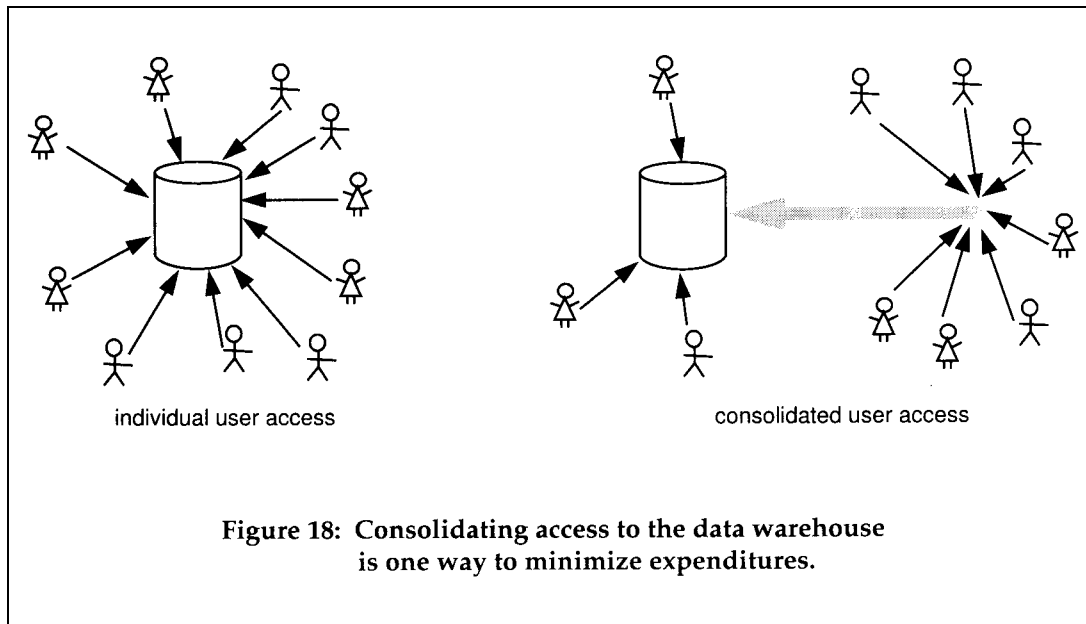


Figure 18: Consolidating access to the data warehouse is one way to minimize expenditures.

The consolidated access approach works well where:

- there are many users accessing the data warehouse,
- the access of data is predictable,
- the same data is being accessed by different users, and
- where the need for the return of data is not immediate.

Not all users will have this pattern of access and for this reason there will always be some one or two users that need to have direct access against the data warehouse. But for those users that do fit this pattern of access, creating a consolidated access mechanism can save considerable resources in the access of data.

SUMMARY

The data warehouse budget can be classified in several ways - by recurring and one-time expenditures, and by capital and operational expenditures. The classifications include hardware, software and development expense. The building of the data warehouse is normally done in an iterative fashion, where initial expenditures are made in the first year and successively lower expenditures are made in successive years as the design of the warehouse settles.

Multiple iterative development efforts will be occurring simultaneously over the years. Each development effort requires its own budgeting cycle. The primary factors affecting one-time capital expenses are the selection of the hardware platform (either parallel or client/server) and whether the development is performed manually or in an automated fashion. The primary budgetary factors affecting long-term recurring expenses are whether log tapes are used to trap ongoing changes and whether the maintenance effort has been automated.

THE DATA WAREHOUSE BUDGET

The two most important things that can be done to control the data warehouse budget are to build the warehouse iteratively and to encourage people to use the data warehouse at the highest level of summarization possible. To this end, chargeback can be instigated. In addition, a consolidated access approach can save massive access against the data warehouse.