A Practical Implementation of the Data Base Machine - Teradata DBC/1012

Robert D. Sloan, Consulting Systems Engineer

Teradata Corporation, El Segundo CA.

Abstract

The notion of Data Base machines has been around almost as long as the computer itself.[1] *Most were based on the idea of moving data access logic down closer to the data.*[2] *Of course, most of these never achieved commercial success. This was due, in part, to the rapid advances in general purpose hardware/software products. However, the more significant barrier was overcoming the buyer's perception that these devices were more trouble than they were worth. In order to overcome this resistance price/performance had to be at least an order of magnitude greater than general purpose platforms. Most never accomplished these goals, and very few gained the crucial market acceptance to become commercial products. One of the exceptions is the Teradata DBC/1012 Data Base Computer. This paper discusses how the convergence of multiple technologies made the product possible and how it is used to solve business problems for companies world wide.*

History of the DBC/1012

The DBC/1012 Data Base Computer was conceived in the early 80s as a data base engine for handling large amounts of data. During this time period the relational model was considered to be theoretically very powerful, but there were very few real commercial implementations that could demonstrate its potential.[3] There was a doubt that Relational Data Base Management Systems, (RDBMs), would ever become practical. The relational model was resource intensive and would just be too slow to tackle real business problems. Most users believed that RDBMSs would be relegated, at best, to handling very small data bases in a decision support role. At this same time traditional hierarchical and network DBMSs such as IBM's IMS[4] and Cullinet's IDMS were being pressed to handle larger and larger amounts of data with even greater throughput and response time requirements. Many companies had grandiose plans to create large corporate level shared data base architectures. Very few people believed that the current DBMS products were suitable for this task.[5] They were cumbersome and too inflexible to handle the access requirements for these new strategic systems. A relational DBMS with the speed and data handling capabilities of the existing DBMSs appeared to be the perfect solution.

At the same time a small number of companies were struggling with a category of Data Base applications that had been tagged as Very Large Data Base, (VLDB), applications.[6] Companies like AT&T and Citibank were trying to build strategic applications centered on Data Bases in the 50 to 100 Gigabyte range. These kinds of problems were way out of the design scope of products like IBM's IMS and Cullinet's IDMS. These products both contained restrictions on the maximum allowable data base size. They also provided weak support for data base partitioning. The required data base reorganization processes also become unmanageable for large data bases. These problems were generally considered totally out of the question for the emerging Relational DBMSs. A little company called Teradata disagreed with the predominate opinion at the time and set out to build a Data Base engine that could handle Data Bases up to one Terabyte or one trillion bytes and was based entirely on relational theory.[7]

Meanwhile, there were remarkable developments coming from companies like Intel and Motorola in the area of single chip microprocessors. These new devices could deliver a significant amount of processing power at a fraction of the cost associated with traditional platforms like Mainframes and Mini-computers. Although the current generation of microprocessors at this time were no match for these more traditional platforms when viewed by themselves, there was enormous potential if a large number of these processing elements could be coupled together to make up a large parallel processing complex. The challenge was developing an architecture that could support hundreds of processors tied together to make up a single processing platform.

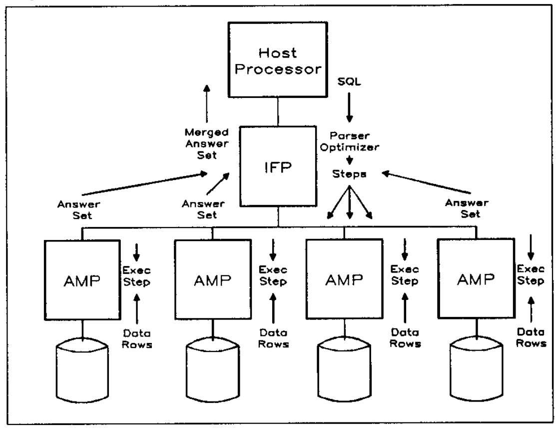
The founders of Teradata recognized the market potential for a relational data base computer that could handle up to a Terabyte (One Trillion Bytes). They also recognized that the way to apply enough processing power to the problem without running into astronomical costs was to use a parallel microprocessor architecture. On this basis, Teradata Corporation was founded with the objective of developing the DBC/1012.

Solving the Parallel Processor Problem

The idea of parallel processing has been around for a long time. Many different architecture topologies had been developed. Many of them were based on a limited number of moderately powered processors sharing a common memory and controlled by a single copy of the operating system. The performance of these systems was degraded by access contention for the common shared memory. Thus the performance characteristics were non-linear. If one processor had a power of 1.0 then two processors were 1.8 and three were 2.7, and so forth. When you reached 6 or 8 processors there was little or no advantage to adding an additional processor. The specific nature of the bottleneck varies from one implementation to the next. An example would be multiple processors sharing a common system bus with main memory located on the bus. The degradation is caused by the bus arbitration hardware forcing other processors to wait while one processor fetches or stores data into the common memory. As more processors are added the probability of bus collisions increases to the point where an additional processor would offer no performance gains because it would not be allowed to gain access to the memory bus. The biggest problem, however, actually resided with the software. The traditional DBMS software architectures were not designed to take advantage of a multi-processor architecture. The data access routines had not been broken up into units of work that could be done in parallel. Software patches were mostly oriented to increasing throughput by allowing multiple copies of key tasks to be dispatched in parallel. It was evident that a whole new DBMS architecture would be required to truly exploit the massively parallel processing platform.

To solve this problem a Client/Server architecture was adopted. Client Processors would accept the request from an attached host processor, parse the requests, build an execution plan, and then dispatch the requests to multiple Data Base Servers as needed. In order to allow as many Data Base Servers as possible to work on the same problem a new data distribution technique would be required. The rows associated with a data table are evenly distributed across all the Data Base Servers in the processing complex. Each Data Base Server has its own disk storage devices that are managed independently of other Data Base Servers in the complex. All communications between Clients and Servers is accomplished via messages that are transmitted on a high speed network that interconnects all the processors in the complex. This "divide and conquer" approach to complex data base queries, is the heart of the DBC/1012.[8] See Figure 1 for a diagram of the DBC/1012 Client/Server message flow.

Figure 1 - DBC/1012 Client/Server Data Flow

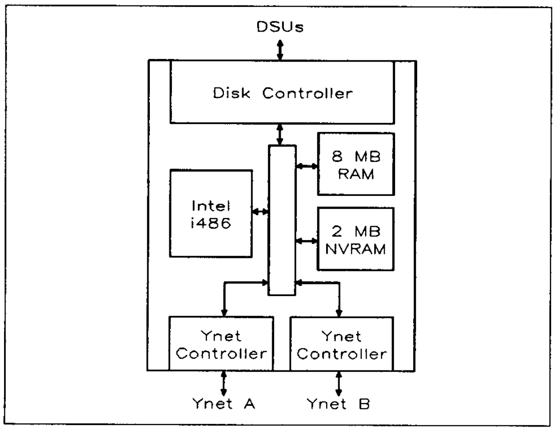


The DBC/1012 Architecture

The DBC/1012 is made up of three types of processor modules each handling a specific task in the architecture:

- AMPs or Access Module Processors perform the actual DBMS functions. Each AMP has one or more of its own direct access storage devices, known as Data Storage Units or DSUs in DBC/1012 terminology. The data base tables are evenly divided among the AMPs on the system using a hashing algorithm that is applied to the primary index for the table. With the Fallback option a second copy of the data is spread across several other AMPs, so they can take over if the primary AMP fails. The AMPs interpret and execute instructions from one of the two types of interface processors discussed below. These instructions are known as "Steps" and they provide the AMPs with the information needed to access the data stored on the DSUs under its control. Once the AMPs have received the Steps to execute, they operate completely independently of the other AMPs in the processor complex. A typical DBC/1012 system will have from 4 to 200 AMPs. The maximum number of AMPs in a single DBC/1012 is around 900. See Figure 2 on the following page for a digram of the AMP.
- IFPs or Interface Processors perform two basic functions. They provide channel connectivity to a

Figure 2 - DBC/1012 AMP Processor Module



Mainframe type host computer. They also are responsible for parsing the SQL, (Structured Query Language), requests and generating the optimal query execution plan and then coordinating its execution on the AMPs. The IFP is also responsible for establishing user sessions with the DBC/1012 and load balancing those sessions among the IFPs connected to a particular host so that each IFP is handling an equal amount of the host workload.

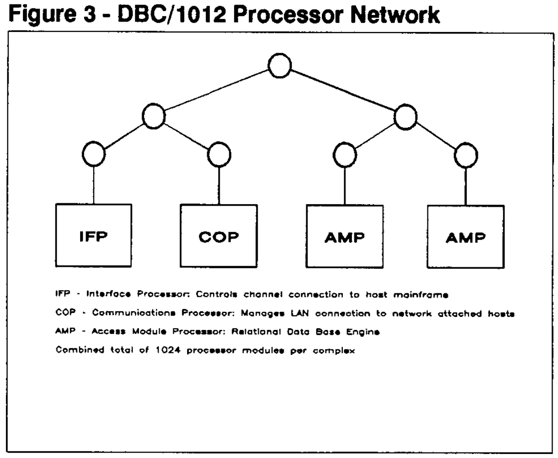
- COPs or Communications Processors are very similar to IFPs except that they are designed to provide a connection to a LAN based host or workstation instead of a mainframe host. COPs and IFPs both have the ability to store the execution plan associated with previously parsed requests in a Statement Cache. When an SQL request is received from an attached host, the Statement Cache is checked to see if the SQL requests has already been parsed. If it has, the Steps are fetched from the Cache and the Parsing/Optimizer phase can be skipped. This allows for rapid execution of repetitive requests, such as in an OLTP, (On-line Transaction Processing), environment.

All the processors in the system are interconnected via a pair of high speed Buses known as Ynets. See Figure 3 for a diagram of the DBC/1012 interconnected processor network. The Ynet(tm) is a proprietary 6 Megabyte per second BUS developed by Teradata specifically for the DBC/1012. The architecture of the Ynet resembles a diagram of a tournament. All processors are connected to nodes at the lowest level in the network. A set of nodes one level up interconnect the lowest level nodes. Another set of nodes interconnect the next level of nodes, and so on, until you get to a single node. These interconnected nodes look like upside down Ys, hence the name Ynet. The Ynet is highly oriented to message broadcasting, message passing, and most importantly sorting. The Ynet is used to dynamically sort answer sets as they are being transmitted from the AMPs to the requesting IFP or COP. All these factors are crucial to the performance characteristics of the DBC/1012. The functions that are performed by the Ynet are largely responsible for the highly linear processing characteristics of the DBC/1012. For example if a DBC/1012 with 24 AMPs can process a full table scan query in 4 minutes, a DBC/1012 with 48 AMPs would process the same query in 2 minutes.

The actual processor modules are made up of a main Intel i486(tm) processor with 8 MBs of 32 bit RAM, (Random Access Memory), for the Model 4 Processor Module or an Intel i386(tm) with 4 or 8 MBs of 32 bit RAM for the Model 3. Each processor module contains an intelligent interface controller for each of the two Ynets. The AMPs also contain an intelligent SMD, (Small Peripheral Device Controller), or IPI, (Intelligent Peripheral Interface), disk controller that can drive up to 4 Seagate Sabre5(tm) 1.2 GB disk drives or Seagate Sabre6(tm) 2.5 GB disk drives. A Non-volatile RAM option, called NVRAM in Teradata terminology, can also be added to each AMP. This feature allows safe writes to a special memory module that is backed up by a battery. In the event of a system failure the data can be reliably recovered from the NVRAM module. This feature provides significantly higher performance for update type transactions. As soon as buffers are written into NVRAM the "End Transaction" can be processed thus allowing locks to be released and control returned to the user much faster than if the system had to wait for disk writes.

The IFP processor modules also contain a Channel Interface Controller or CIC that can connect to a standard IBM compatible Channel or other mainframes depending on the type of CIC. Instead of the CIC the COP contains an intelligent Ethernet Adapter that allows for attachment to Local Area Networks, (LANs), for

Figure 3 - DBC/1012 Processor Network



TCP/IP, (Transmission Control Protocol/Internet Protocol), communications with attached hosts and workstations.

The DBC/1012 is packaged into processor cabinets that can house up to 8 processors modules of the AMP, IFP, or COP variety. Storage cabinets are designed to house up to 16 Seagate Sabre5 or Sabre6 DSUs. The cabinets are equipped with an AC Power Distribution Unit or PDU and individual DC power supplies for each component in the cabinet. With the Fallback data storage option, up to one AMP can fail in each of the Fallback clusters and the DBC/1012 can continue to operate and provide data access to its users in a somewhat degraded performance mode. Fallback clusters can be varied in size but are usually set at around 4 AMPs per cluster.

The software for the DBC/1012 consists of several functionally specific components. The software that runs on the actual server is made up of two components:

- **DBS (Data Base System)** - The DBS software was originally written in mostly PASCAL with some Intel Assembler language routines. It includes Session Management, the Parser, Optimizer, and Query Execution control routines which run on the IFPs and COPs and Data Base access routines which run on the AMPs.
- **TOS - Teradata Operating System** - Consists of low level control routines that deal with memory management, task control, dispatching, error recover, device control, and message passing. Teradata decided to develop the TOS operating system, because in the early 80's when the DBC/1012 was being developed there were no suitable virtual storage operating systems available for the Intel 80x86 microprocessor. While having to develop the unique operating system added complexity to the project, it also provided flexibility. Teradata was able to optimize TOS for DBMS type processing and the parallel architecture of the hardware.

The software that runs on each host platform is considerably simpler than the DBS or TOS software and is made up of two basic components:

- **API (Application Program Interface)** - There are two methods provided for an applications program to interface with the DBC/1012. You can imbed the SQL into your Cobol or "C" language program and run it through the SQL Host Preprocessor utility or you can include calls to library subroutines using the Call Level Interface or CLI. In either case the interface software is responsible only for placing the SQL into a message called a Parcel and sending it to the DBC/1012 interface software component and waiting for the answer set to be returned in Parcels.
- **TDP (Teradata Director Program)** - This software component is responsible for establishing and maintaining communications between the host computer and the DBC/1012. The TDP will accept parsels from the Call Level Interface or Preprocessor programs and sends them to the DBC/1012 over the communications media. When answer set Parcels are returned it directs them back to the appropriate applications program. The TDP for LAN attached hosts is called the Micro Teradata Director Program or MTDP.

In addition to the basic host interface software a set of utilities is also provided for each host platform. These include:

- **Fastload** - High speed load utility for populating new tables. The Fastload utility supports programming exits that allow extracts from other DBMS environments to be overlapped with the load process.
- **Dump/Restore** - Dump all or a portion of a DBC/1012 data base to a storage device using an attached host processor.
- **BTEQ (Batch Teradata Query Utility)** - Submit a batch SQL query. BTEQ can be used to create batch reports, perform data base maintenance functions and test SQL queries.
- **Bulkload and Multiload** - perform batch updates against existing data tables. Bulkload serves as a batch driver to send SQL update transactions to the DBC/1012 using multiple simultaneous sessions to exploit its parallel processing capability. Multiload is a new utility that has just been developed to support high performance batch updates. Multiload is supported by new DBS code that processes the batch SQL using techniques that reduce communication with the host processor, reduce IFP and AMP CPU, and reduce data base I/O.

In addition Teradata has formed relationships with several third party vendors to provide a wide variety of tools for use with the DBC/1012. These include transaction monitors, forth generation languages, natural language interfaces, report writers, and complete application development environments.

Performance Characteristics

The DBC/1012 is probably best known for its ability to perform complex queries against large data bases. For example a data table with 1 billion rows stored on a 200 AMP DBC/1012 would be divided into 200 separate partitions such that each AMP would be responsible for 5 million rows. If each row were 50 characters long then the data would occupy about 39,060 blocks on each AMP. At

40 I/Os per second per AMP it would take under 17 minutes to do a full table scan type query. This is possible because a 200 AMP DBC/1012 can sustain I/O rates of 8,000 I/Os, or more, per second for a single query with total aggregate I/O rates as high as 17,000 I/Os per second. This type of query is obviously a worse case scenario and response times could be reduced by an order of magnitude if indexes had been available for the columns that are being qualified in the query. The point is that this is a very realistic situation in a decision support AD HOC query environment. It is impossible to predict every possible access path into a table. Even if one knew all the access paths, it is unlikely that the user could afford the disk space and maintenance overhead to create every possible secondary index that might be required. Under most RDBMS architectures this type of query on tables much smaller might take hours or even days to complete and thus would be totally impractical to run.

The key to the complex query performance advantage of the DBC/1012 lies in two specific areas:

- The ability to sustain high I/O rates due to the data distribution techniques and parallel architecture of the DBC/1012. Each AMP has its own I/O controller, buffer pool, and processor and thus can operate independently of all the other AMPs in the processor complex.
- The DBC/1012 is constructed from mass produced single chip processor elements. This allows processing power to be delivered at a cost of around $5,000 per MIP. This is a significant advantage when compared to mainframe processing power that cost $50,000, or more, per MIP. This allows the massive amounts of processing power to be applied to a single problem at cost levels that are still economically feasible.

The DBC/1012 architecture also excels with the exact opposite sort of workload. Simple "prime key" type requests are directed by the interface processors to a single AMP that can process the entire request independently of the other processors in the complex. This yields almost perfectly linear processing characteristics over a range of throughputs up to 1000 plus transactions per second.

Another important performance aspect of the DBC/1012 is its ability to load data very rapidly. This is very important in environments where data is being extracted from an operational DBMS and loaded into an AD HOC support platform to facilitate time critical decision support applications. The key to this performance advantage is the Fastload utility coupled with the parallel architecture of the DBC/1012. The first phase of the Fastload utility downloads the data from the host to the DBC/1012 at speeds that are most commonly limited by the speed of the channel or the Fastload input device. The second phase of the Fastload operates in a highly parallel manner. Each AMP takes its portion of the input data and reformats it into the internal row format, sorts it into the proper hash code sequence, and then sends it to the appropriate AMP to be stored.

The bottom line in all these scenarios is price/performance and more importantly scalability. As data volumes or workloads grow, AMPs and DSUs can be added to provide a smooth and orderly transition to the processing levels required. The DBC/1012 includes built in utilities that can redistribute the data as AMPs and DSUs are added. These utilities operate completely independent of the attached hosts. These feature make the DBC/1012 highly suitable for applications where requirements are unpredictable and subject to substantial changes especially on short notice.

Features of the DBC/1012

Besides the performance and scalability features mentioned previously there are several other significant advantages offered by the architecture of the DBC/1012. For example:

- The multi-host connectivity features of the DBC/1012 allows it to be connected to mainframes, mini-computers, workstations, and personal computers, providing a flexible shared data base environment. This provides a foundation that leverages the relative strengths of the available host platforms. Mainframes can be used for batch and tape intensive operations. Workstations and personal computers can be used to provide a powerful low cost presentation layer for user access. Low cost "open systems" based platforms can be used as process servers for the workstations. Since as many hosts as needed can be connected to the DBC/1012, functions can be spread across multiple low cost hosts without worrying about degradation when accessing the data base. These architecture features are also very useful for migration from one host platform to another. Applications can be transitioned from one platform to another in an orderly manner on a function by function basis thus avoiding the high cost, high risk, and business disruptions associated with flash-cut type rewrites.
- The file system of the DBC/1012 completely manages the placement of all data on the system. The tables are dynamically reorganized on an ongoing basis to insure optimal data placement without the use of periodic batch data base reorganization utilities. This is especially critical on large systems that can have more than a 1000 DSUs. The space allocated to a data base

can be increased or decreased dynamically through a simple command in seconds without unloading and reloading the data.

Application of the DBC/1012

The DBC/1012 is used in a wide variety of applications where high data volumes are coupled with complex data analysis requirements. See Figure 4 for a partial customer list by industry. These range from retail applications involving the analysis of "point of sale" data, long distance carriers analyzing calling patterns, network utilization, and performing revenue forecasting, to insurance companies doing complex analysis of claim data. Most of these applications represent breakthrough type of business processes. They were applications that generally were not feasible before the development of the DBC/1012. They would have been so slow as to not be useful or too expensive using the existing mainframe based DBMSs that were available prior to the DBC/1012.

See Figure 5 on the following page for a sample list of applications that have been built on the DBC/1012.

Other classes of applications include state of the art client/server based applications utilizing high end workstations to provide powerful Graphical User Interface, (GUI) based functions. These applications represent the leading edge of the current trend in downsizing applications from the traditional mainframe architectures to LAN based client/server architectures that utilize cost effective open systems based platforms. For example, powerful tools available on an X-Windows based UNIX$_{(tm)}$ workstation can be employed to solve business problems using data bases of 100 Gigabytes or more.

Another option available is the ability to implement high transaction throughput applications utilizing UNIX processing platforms. Without the DBC/1012 the architect was faced with the problem of trying to fit the application processes and the DBMS on a single processor to avoid the networking overhead associated with distributed transaction processing. Implementing a large

Figure 4 - Partial Customer List by Industry

Aerospace and Manufacturing
- Boeing
- Harris Corp
- Karsten
- McDonnel Douglas
- Northrop
- Xerox Canada

Banking and Finance
- ANZ Bank (Australia)
- Bank of America
- Chemical Bank
- Citibank
- Commerzbank (Germany)
- Gerling-Konzern (Germany)
- Kemper Service Company
- Merrill Lynch
- TSB Trust Company (U.K.)

Communications
- AT&T
- Bell Atlantic
- Bell South
- British Telecom PLC (U.K.)
- GTE
- Televerket (Sweden)
- US Sprint
- US West

Government
- Department of Supply and Service (Canada)
- Internal Revenue Service
- KDO (Norway)
- Stockholm City Data Services (Sweden)
- Teale Data Center (State of California)
- U.S. Government

Insurance
- CNA
- Guaranty National Insurance
- Liberty Mutual
- Prudential Insurance (U.K.) Ltd.
- Royal Insurance (U.K.) Ltd.
- Transamerica

Pharmaceutical and Chemical
- Baxter Laboratories
- Chevon
- Ciba-Geigy (Switzerland)
- Shering Plough

Retail and Consumer
- A-Lehdet Oy (Finland)
- Great Universal Stores (U.K.)
- Kmart
- Kraft
- Helene Curtis
- Littlewoods (U.K.)
- Mervyn's
- Otto Versand (Germany)
- Pripps Brewery AB (Sweden)
- Proctor and Gamble
- QVC

Transportation
- American Airlines
- American Presidents Companies

Other
- Brigham Young University
- Kansas City Power and Light
- Ontario Hydro
- Owens Corning

Figure 5 - Sample DBC/1012 Applications

Industry	Application	COPs/IFPs x AMPs x DSUs
Communciations	*Financial Analysis*	*0/20 x 192 x 192, 386, 1.2GB*
Retail	*Purchasing/Sales Analysis*	*2/12 x 42 x 42, 386, 1.2GB*
Communications	*Service Provisioning*	*29/60 x 92 x 184, 486, NVRAM, 1.2GB*
Retail	*Purchasing/Sales Analysis*	*0/12 x 52 x 104, 486, 2.5GB*
Insurance	*Actuarial Analysis*	*5/12 x 112 x 448, 386, 1.2GB*
Retail	*Purchasing/Sales Analysis*	*6/40 x 244 x 488, 386, NVRAM, 1.2GB*
Communications	*Long Distance Traffic Analysis*	*1/12 x 96 x 192 486, 2.5GB*
Retail	*Purchasing/Sales Analysis*	*2/5' x 80 x 80, 386, 1.2GB*
Communications	*Network Administration*	*18/6 x 52 x 52, 386, 1.2GB*

data base on a UNIX platform was also inherently plagued by shortcomings in the area of tape handling associated with backup and recovery functions. With the DBC/1012 multiple UNIX front end hosts can be connected to a single DBC/1012 to provide a high performance, cost effective UNIX OLTP environment.

One of the more strategic uses of the DBC/1012 involves architectures where a single shared data base will be used to satisfy a wide variety of highly diverse business processes that all have need to access a single copy of data. These architectures rely on all the major strengths of the DBC/1012:

- Multi-platform connectivity is needed to support the variety and number of host platforms that will be required to satisfy all the processing requirements. This includes strategic platforms in addition to legacy platforms that must be connected to facilitate migration.
- High throughput capacity is needed to sustain the transaction rates resulting from many diverse business processes all running off a single physical copy of data. The nature of new more demanding strategic systems may also require higher throughput capacities.
- The flexibility of the DBC/1012 is needed to support the varied number of access paths that are required to support all the known functions, as well as, functions that will be added in the future. This includes the ability to add and drop columns in data tables, the ability to add and drop secondary indexes. The flexibility also extends into day to day administration features like altering security specifications. It includes the DBC/1012s use of dynamic cached requests so the user is freed from the need to build and maintain query plans using utilities.
- The scalability features of the DBC/1012 are needed to allow for quick reaction to new load requirements and data storage requirements that result when new business processes are added or when the characteristics of existing business process change. This also includes scalability in terms of adding more host processors while still having access to a single common physical data base.
- The large data base handling characteristics of the DBC/1012 are important in dealing with the need to store massive amounts of detail data in order to satisfy the information granularity requirements of todays strategic business systems.

Future of the DBC/1012

On April 22, 1991 Teradata Corporation introduced the DBC/1012 Model 4, the fourth generation of the company's data base computer system. The Model 4, based on Intel's 33 Mhz i486 microprocessor, greatly extends Teradata's undisputed advantage in large-scale decision support and provides significant new capabilities in production processing environments. The DBC/1012 will continue to grow in processing power and capability as it rides the technology wave of the single chip microprocessor. Intel i486 microprocessors are expected to be increased from the current 33 Mhz clock speed to at least 66 Mhz, with some analysts predicting 100 Mhz clock speeds. The Intel i586(tm) will most certainly top even these speeds. This provides a CPU boost for DBC components in the range of 100 to 200 %. Disk technology is expected to yield high capacity 2.5 to 5 Gigabyte drives in form factors of 5.25 inch as opposed to the 8 inch 1.2 and 2.5 Gigabyte drives currently deployed in the DBC/1012. In addition the move to Disk Arrays, where multiple smaller disks are used to emulate a larger disk, will improve disk economies, as well as, possibly reduce the need for Fallback in the future. These advances coupled with feature enhancements in the DBC/1012 software will yield future capabilities for data bases up to 10 Terabytes or more with even greater throughput and reliability.

However, the much more exciting aspect to the future of the Teradata DBC/1012 is the vast number of new applications that Teradata's customers can and will dream up. As more and more users see the possibilities of such a large high performance data base server that can be connected to almost any platform, we will see more and more new application ideas come to fruition. An increasing number of companies are discovering the huge competitive advantage offered by allowing key decision makers to have direct access to the vast quantities of detail data that is associated with most industries. Retail buyers can make better decisions if they have access to store level "Point of Sale" data. Long distance marketing people can make better decisions if they have access to long distance calling trends by demographic groups, etc.

Bibliography

[1] International Workshop on Data Base Machines, Munich 1983. Edited by H. O. Leilich & M. Missikoff

[2] DeWitt, D. J. and Hawthorn, P. B., "A Performance Evaluation of Data Base Machine Architectures," Seventh International Conference on Very Large Data Bases, September, 1981

[3] Judy M. King, "Evaluating Data Base Management Systems," (New York: Van Nostrand, 1981)

[4] Information Management System Virtual Storage (IMS/VS), General Information Manual, GH20-1260, IBM, White Plains, N.Y., 1981

[5] E. F. Codd,"Derivability, Redundancy, and Consistency of Relations Stored in Large Data Banks," IBM Research Report RJ599 (August 1969)

[6] Proceedings, 9th International Conference on Very Large Data Bases, Florence, Italy (November, 1983)

[7] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," CACM 13, No 5 (June 1970)

[8] DBC/1012 Concepts and Facilties, C02-0001-05, Teradata Corp, El Segundo, CA, 1990