**Teradata Architecture**

**Introduction to Teradata RDBMS**

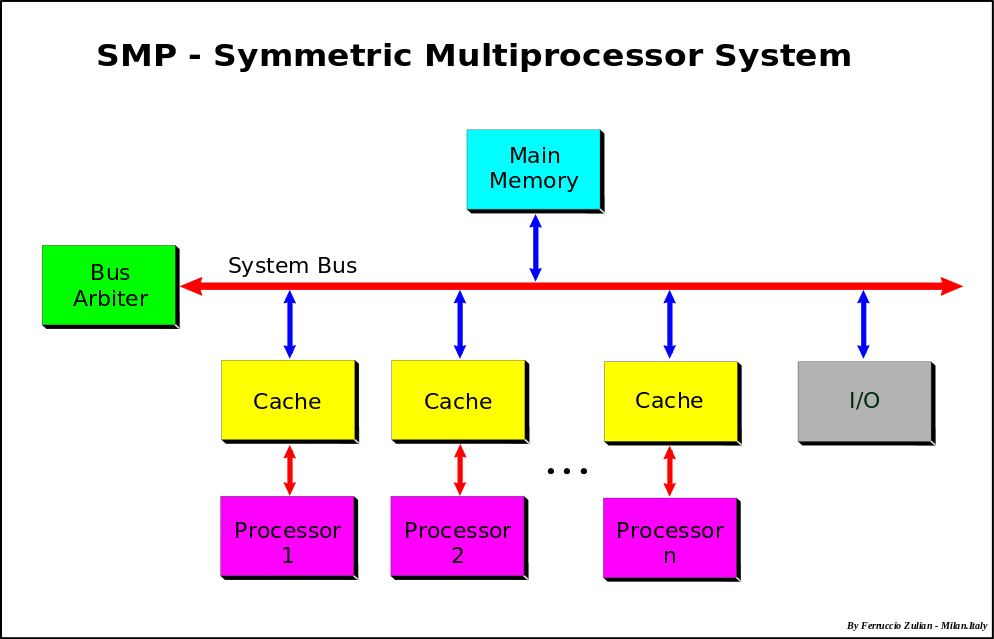
Teradata RDBMS is a complete relational database management system. The system is based on off-the-shelf Symmetric Multiprocessing (SMP) technology combined with a communication network connecting the SMP systems to form a Massively Parallel Processing (MMP) system.

**BYNET** is a hardware inter-processor network to link SMP nodes. All processors in a same SMP node are connected by a virtual BYNET.

**What is symmetric multiprocessing SMP?**

Symmetric multiprocessing (SMP) involves a [multiprocessor](https://en.wikipedia.org/wiki/Multiprocessor) computer hardware and software architecture where two or more identical processors are connected to a single, shared [main memory](https://en.wikipedia.org/wiki/Main_memory), have full access to all I/O devices, and are controlled by a single operating system that treats all processors equally, reserving none for special purposes. Most multiprocessor systems today use an SMP architecture. In the case of [multi-core processors](https://en.wikipedia.org/wiki/Multi-core_processor), the SMP architecture applies to the cores, treating them as separate processors.

SMP systems are [tightly coupled multiprocessor](https://en.wikipedia.org/wiki/Multiprocessing#Processor_coupling) systems with a pool of identical processors running independently of each other. Each processor, executing different programs and working on different sets of data, has the capability of sharing common resources (memory, I/O device, interrupt system and so on) that are connected using a [system bus](https://en.wikipedia.org/wiki/System_bus).



**Teradata** architecture is based on **Massively Parallel Processing (MPP)** architecture. The biggest strength of the Teradata is the parallelism. So the architecture of the Teradata is designed in such a way to keep this strength in mind. Teradata is unique from any other database because of its unique architecture only.

NCR Licensed AT&T UNIX SVR4 (AT&T owned UNIX in those days) in 1989 and modified it for their use.

NCR added code to enhance **Reliability, Availability and Serviceability** **(RAS)** and **Multiprocessing** (**MP**) to SVR4.0, and that's why it's called UNIX MP-RAS. It was one of the first UNIX multiprocessor OS.

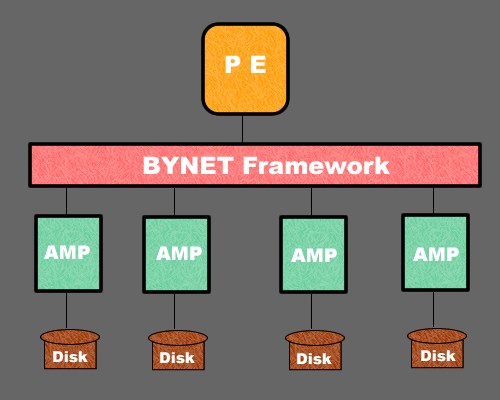
To make Teradata run on **UNIX MP-RAS**, the Teradata developers wrote extension code on top of UNIX MP- RAS to give it the features they needed and the compatibility with the Teradata RDBMS and it is called as **Parallel Database Extensions (PDE).**

**PDE was added to the UNIX kernel by NCR to support the parallel software environment.**

The major components of Teradata are **Node,** **Parsing Engine**, **BYNET and Access Module Processors (AMPs).**

We look into these components in details after looking the logical view of the architecture.

The logical view of Teradata architecture is given below –



**Node** − It is the basic unit in Teradata System. Each individual server in a Teradata system is referred as a Node. A node consists of its own operating system, CPU, memory, own copy of Teradata RDBMS software and disk space. A cabinet consists of one or more Nodes.

**Parsing Engine (PE)** – This component is an interface layer on the top of operating system. Whenever a user login to Teradata it actually connect to Parsing Engine (PE). When a user submits a query, then the PE takes action, it creates a plan and instructs AMPs what to do in order to get the result from the query. The PE knows all, it knows how many AMPs are connected to Teradata system, how many rows are in the table and what is the best possible plan to execute the query. This is why the PE is also called as the ‘**OPTIMIZER**’.

Besides making a perfect plan for query execution PE also makes a check on the access right of the user that weather the user has the privilege to execute the query or not. In this way PE also perform security feature implementation on the users.

**Parsing Engine (PE)** is a type of vproc that has software components to break SQL into steps, and send the steps to the AMPs.

1. **Session Control** - When you log on to the Teradata RDBMS through your application, the session control software on the PE establishes that session. Session control also manages and terminates sessions on the PE.
2. **Parser/Optimizer** - The parser interprets your Teradata SQL request and checks the syntax. The parser decomposes the request into AMP steps, using the optimizer to determine the most efficient way to access the data on the virtual disks. Then the parser sends the steps to the dispatcher.
3. **Dispatcher** - The dispatcher is responsible for a number of tasks, depending on the operation it is performing:
4. Processing Requests
5. Processing Responses

The responsibilities of parsing engine are −

* Communicate with client
* Manage sessions
* Receive the SQL query from the client
* Parse the SQL query check for syntax errors
* Perform semantic check (Objects and Privileges checking)
* Prepare the execution plan to execute the SQL query and pass it to BYNET
* Receives the results from the AMPs and send to the client

**Access Module Processor (AMP)** – AMP, acronym for "Access Module Processor," is the type of vproc (Virtual Processor) used to stores and retrieves the data, manage the database, handle file tasks and manipulate the disk. Each AMP attached to the Teradata system listens to the PE via the BYNET for instructions. Each AMP is connected to its own disk and has the privilege to read or write the data to its disk. The AMP can be best considered as the computer processor with its own disk attached to it. Whenever it receives the instructions from the PE it fetches the data from its disk and sends it back to PE through BYNET. Each AMP is allowed to read and write in its own disk ONLY. This is known as the ‘SHARED NOTHING ARCHITECTURE’. Teradata spreads the rows of the table evenly across all the AMPs, when PE asks for data all AMPs work simultaneously and read the records from its own DISK. Hence a query will be as slow as the slowest AMP in the system. This is known as parallelism.

When AMPs receive the data and execution plan from PE, performs if any data type conversion, aggregation, filter, sorting and stores the data in the disks associated with them.

When the client runs queries to retrieve records, the Parsing engine sends a request to BYNET. BYNET sends the retrieval request to appropriate AMPs. Then AMPs search their disks in parallel and identify the required records and sends to BYNET. BYNET then sends the records to Parsing Engine which in turn will send to the client.

In Teradata V1, the Access Module Processor (AMP) was the physical processing unit for all the Teradata database functions. Each AMP then contained its own microprocessor, disk drive, file system, database software (Database Manager), Teradata Operating System (TOS), and YNET interface. In that sense, each AMP was a node.

In Teradata V2, AMPs become software entities, and thus more flexible units that "deliver basic query parallelism to all work in the system." The number of AMPs (2 - 20 per node) is defined for the Teradata database before the database is loaded.

**FUNCTIONS**

The functions of AMP can be classified as the following:

1. BYNET interface, or Boardless BYNET interface;

2. Database management:

1. Locking;

2. Joining;

3. Sorting;

4. Aggregation;

5. Output data conversion;

6. Disk space management;

7. Accounting;

8. Journaling;

3. Interface to disk subsystem.

4. File-subsystem management;

AMP - The AMP is a type of vproc that has software to manage data:

1. **AMP Worker Task (AWT)** Functions in the AMP perform a number of operations, including:
2. Locking Tables
3. Executing Tables
4. Joining Tables etc. as mentioned above.
5. Executing end transaction steps
6. The file system software accesses the data on the virtual disks. Each AMP uses the file system software to read from and write to the virtual disks.
7. Console Utilities - The AMP software includes utilities to perform generally sophisticated, low-level functions such as:
8. Configure and reconfigure the system
9. Rebuild tables
10. Reveal details about locks and space status

**BYNET** – The BYNET is the communication channel between PE and AMP. It ensures that the communication between PE and AMP is correct and on the right track. In Teradata system there are always two BYNET systems. They are called as ‘BYNET 0’ and ‘BYNET 1’. But we refer them as a single BYNET system. The reason two BYNET exist on a Teradata system is that –

1. If one BYNET fails, the second BYNET takes over the place.
2. Two BYNET improve the performance of the system, the PE and AMP can talk to each other over both BYNET which increase the pace of the communication. It provides a means of load balancing.

Function of BYNET

1. The BYNET is a high-speed interconnect that is responsible for:
2. Sending messages
3. Merging data
4. Sorting answers
5. The BYNET messaging capability enables vprocs to send different types of messages:
6. Point-to-Point - A vproc can send a message to another vproc:
7. In the same node using BYNET software only, the message is reassigned in memory to the target vproc.
8. In another node the message is using both BYNET hardware and software.
9. Multicast - A vproc can send a message to multiple vprocs by sending a broadcast message to all nodes. The BYNET software on the receiving node determines whether a vproc on the node should receive or discard the message.
10. Broadcast - A vproc can broadcast a message to all the vprocs in the system.

**BYNET (Banyan Network)**

It acts like “Message Communication” layer between components.

1. SMP BYNET [PE-AMP]
2. POINT-POINT : One message from PE – One Amp
3. MULTICAST : One message from PE- Many Amps
4. BROADCAST : One message from PE – All Amps
5. MPP BYNET [NODE-NODE]
6. POINT-POINT : One message from One Node to Another Node
7. MULTICAST : One message from One Node to Many Node
8. BROADCAST : One message from One Node to All Nodes

**Short summary –**

The PE checks the syntax of the query, also perform semantic checking

Then PE comes up with the best optimized plan for the execution of the query

The PE passes this plan through BYNET to AMPs.

The AMPs follow the plan and retrieve the data from its DISK.

Then AMPs passes the data to PE through BYNET.

The PE then passes the data to the user.

**Teradata Software Download Link**

[**http://downloads.teradata.com/download/database/teradata-express-for-vmware-player**](http://downloads.teradata.com/download/database/teradata-express-for-vmware-player)

**CLI (Call Level Interface):**

A SQL query is submitted and transferred in CLI packet format

**TDP (Teradata Director Program):**

Route the packets to the specified Teradata RDBMS server

Teradata RDBMS has the following components that support all data communication management:

* Call Level Interface (CLI)
* WinCLI & Open Database Connectivity (**ODBC**)
* Teradata Director Program (TDP for channel attached client)
* Micro TDP (TDP for network attached client)

**Node hardware and software components**

CPUs are not physically associated with vprocs. Performance is best when you use the UNIX affinity scheduler to keep a logical association between a CPU and a vproc.

Memory - Vprocs share a free memory pool within a node. A segment of memory is allocated to a vproc for use, and then returned to the memory pool for use by another vproc.

MCA - Slots in the MCA (Micro Channel Adapter) are used for the following connections:

1. Local Peripheral Board (LPB)
2. External disk arrays
3. LAN connections
4. Mainframe channel connections

MCCA - MCCA boards (Micro Channel Cable Adapter) enable communication between a channel-attached node and the Tailgate box. MCCA boards are located in MCA slots.

Ethernet Card - Each LAN connection to a node requires an Ethernet card, which communicates with the Teradata Gateway software. Ethernet cards are located in MCA slots.

Twisted Pair Shielded Cable - Connects the MCCA card to the Tailgate box for a mainframe channel connection.

LAN Cable - Connect the Ethernet cards in the MCA to the LAN.

Tailgate Box - An adapter between the node cabinet and the mainframe in a channel-connected system.

Bus and Tag Cables - Connects the Tailgate box to the mainframe.

Virtual Disk (vdisk) - The logical disk that is managed by an AMP. Each AMP is associated with a single vdisk.

UNIX - The Teradata RDBMS is built on the UNIX operating system for an open environment. NCR added MP-RAS extensions to UNIX to facilitate a multiple CPU environment.

Parallel Database Extensions (PDE) - Software that runs on UNIX MP-RAS. It was created by NCR to support the parallel environment.

Trusted Parallel Application (TPA) - A TPA uses PDE to implement virtual processors. The Teradata RDBMS for UNIX is classified as a TPA.

Teradata Gateway - Software that communicates between the PEs and applications running on network-attached clients. The Teradata Gateway has a session limit of 600 sessions. There is one Gateway per node.

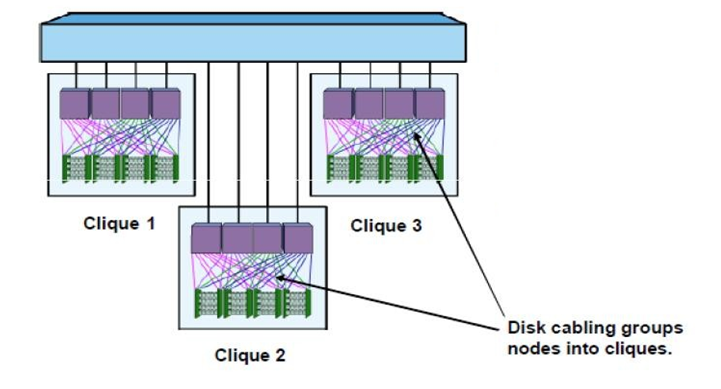
Channel Driver - The Channel Driver software is the means of communication between the application and the PEs assigned to channel-attached clients.

**Teradata Platform:**

1. **Single Node System:** All of the node components together comprise a node. A single node system is typically implemented on Symmetric Multiprocessing (SMP) platform. The vprocs in an SMP system communicate over the vnet.
2. **Multi Nodes System:** Nodes working together create a multiple-node Teradata RDBMS system, which is implemented on Massively Parallel Processing (MMP) system. The nodes and vprocs communicate over the BYNET (Banyan Network).

**Clique**

1. A clique is a set of Teradata nodes that share a common set of disk arrays which are connected in daisy chain network to each disk array controller.
2. Cliques provide high data availability if a node fails for any reason (i.e. UNIX reset).
3. Vprocs are distributed across all nodes in the system. Each multi-node system has at least one clique.



**FLOW OF SQL STATEMENT**

1. A user generates an SQL query on the channel attached client. The query can either be from a BTEQ session at an interactive terminal, from a compatible fourth generation language, or can originate from within an application program coded in a host language.
2. The CLI request handler packages the request and sends it to the Teradata Director Program (TDP) for routing to the server.
3. The TDP establishes a session, then routes the request across the communications channel to the parsing engine (PE).
4. The parser component of the PE opens the request package and parses the SPL code for processing, interprets it, checks its syntax, and optimizes the access plan.
   * 1. Without errors - The parser decodes the request into a series of work steps and passes them to the dispatcher.
     2. With errors - The dispatcher receives the appropriate error message and returns it to the requester. Processing terminates.

The dispatcher sequences the steps and passes them to the BYNET with instructions about whether the steps are for one Access Module Process (AMP), for an AMP group, or for all AMPs.

1. The BYNET (or virtual BYNET on a single node system) distributes the execution steps to the appropriate AMP for processing.
2. The AMPs process the execution steps by performing operations on the database. The AMPs make these operations by making calls to the file system.
3. The file system performs primitive physical data block operations by locating the data blocks to be manipulated and then passing control to the disk subsystem.
4. The disk subsystem retrieves the requested blocks for the file system.
5. The disk manager returns the requested blocks to the file system.
6. The file system returns the requested data to the database manager.
7. The database manager sends a message back to the dispatcher stating that the data is ready to be returned to the requesting user, then sorts and transmits the data to the interface engine over the BYNET.
8. The BYNET merges the sorted response and returns it to the requisition interface engine for packaging.
9. The dispatcher builds the response message and routes it to the requesting client system.
10. The TDP receives and unpacks the response messages and makes them available to the CLI.
11. CLI passes the received data back to the requesting application in blocks.
12. The requesting application receives the response data in the form of a relational table.