Craig S. Mullins

Return to Home Page

Spring 1995

The open enterprise survival guide

An Introduction to the Architecture of DB2

By Craig S. Mullins

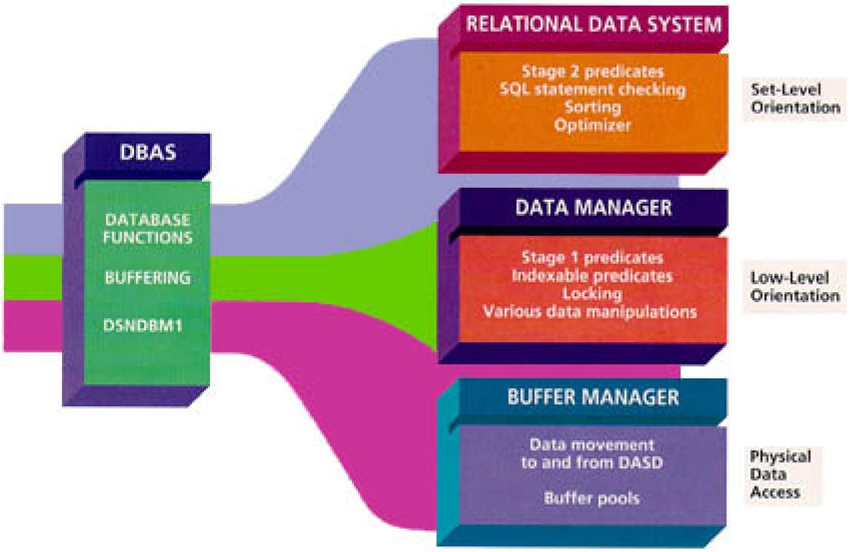
Editor's note: As more sites establish multiple DBMS environments, developers and analysts need to understand the strengths and weaknesses of various client/server DBMS offerings. This article is the second in a series on "Client/Server DBMS Architectures;" here Craig S. Mullins examines the architectural "guts" of DB2. It is adapted from the second edition of his book, DB2 Developer's Guide.

After working with DB2 for MVS for awhile, most IT professionals will have a sound understanding of the fundamental concepts of the DBMS. They will be familiar with the functionality and nature of SQL and know how to embed SQL in an application program. But what is actually going on in DB2 behind the scenes? This article provides a basic introduction to the architecture of DB2, as well as information about each component of that architecture.

Architected around the address space
Conceptually, DB2 is a relational database management system. Physically, DB2 is an amalgamation of address spaces and intersystem communication links that, when adequately tied together, provide the services of a relational database management system.

Beginning with DB2 Version 3, each DB2 subsystem consists of three or four tasks started from the operator console 1. Each task runs in a portion of the CPU called an address space. Version 4 of DB2 provides an additional address space for stored procedures. A description of these five address spaces follows.

The DBAS, or **Database Services Address Space**, provides the facility for manipulating DB2 data structures. The default name for this address space is DSNDBM1, but each individual shop may rename any of the DB2 address spaces. The DBAS is responsible for running SQL statements and managing data buffers. It contains the core logic of the database management system. Three individual components make up the DBAS: the Relational Data System, the Data Manager, and the Buffer Manager. Each of these components perform specific tasks.



Components of the Database Services Address Space

The SSAS, or **System Services Address Space**, coordinates the attachment of DB2 to other subsystems (CICS, IMS/DC, or TSO). SSAS is also responsible for all logging activities (physical logging, log archival, and BSDS). DSNMSTR is the default name for this address space.

The third address space required by DB2 is the IRLM, or **Intersystem Resource Lock Manager**. The IRLM is responsible for managing DB2 locks (including deadlock detection). The default name of this address space is IRLMPROC.

The fourth DB2 Version 3 address space, DDF, or **Distributed Data Facility**, is the only optional one. The DDF is required only if distributed database functionality is required.

The newest address space, SPAS, or **Stored Procedure Address Space**, has been added to DB2 Version 4 to support stored procedures and remote procedure calls (RPC's). The SPAS runs as an allied address space providing an independent environment for stored procedures to execute. This effectively isolates the user-written stored procedure code in its own little world so that it cannot interfere with the system code of DB2 itself.

These five address spaces contain the logic to effectively handle all DB2 functionality.

The functionality of the DBAS

Recall that the DBAS is responsible for executing SQL and is composed of three distinct components. Each component passes a SQL statement to the next component, and when results are returned, each component passes the results back. Let's examine each of these components.

The **Relational Data System** (RDS) is the component that gives DB2 its set orientation. When a SQL statement requesting a set of columns and rows is passed to the RDS, it determines the best mechanism for satisfying the request. Note that the RDS can parse a SQL statement and determine its needs. These needs may include any of the features supported by a relational database (such as selection, projection, or join).

When a SQL statement is received by the RDS, it checks authorization, translates the data element names being accessed into internal identifiers, checks the syntax of the SQL, and optimizes the SQL creating an access path.

The RDS then passes the optimized SQL statement to the **Data Manager** (DM) component. The DM delves deeper into the data being accessed. In other words, the DM is the component of DB2 that analyzes rows (either table rows or index rows) of data. The DM analyzes the request for data, and then calls the Buffer Manager to satisfy the request.

The **Buffer Manager** (BM) accesses data for other DB2 components. A data buffer is often referred to as a cache in other DBMSs. The BM uses pools of memory set aside for the storage of frequently accessed data in order to create an efficient data access environment.

When a request is passed to the BM, it must determine whether the data is in the appropriate buffer pool. If it is, the BM accesses the data and sends it to the DM. If the data is not in the buffer pool, it calls the VSAM Media Manager, which reads the data and sends it back to the BM, which in turn sends it back to the DM.

The DM receives the data passed to it by the BM and applies as many predicates as possible to reduce the answer set. Only Stage 1 predicates are applied in the

DM.

Finally, the RDS receives the data from the DM. All Stage 2 predicates are applied, the necessary sorting is performed, and the results are returned to the requester.

An understanding of the internal components of DB2 can be helpful when developing a DB2 application. For example, consider Stage 1 and Stage 2 predictions. It is easier to understand that Stage 1 predicates are more efficient than Stage 2 predicates because you know that they are evaluated earlier in the process (in the DM instead of the RDS). They therefore avoid the overhead associated with the passing of additional data from one component to another.

Additional architectural concerns

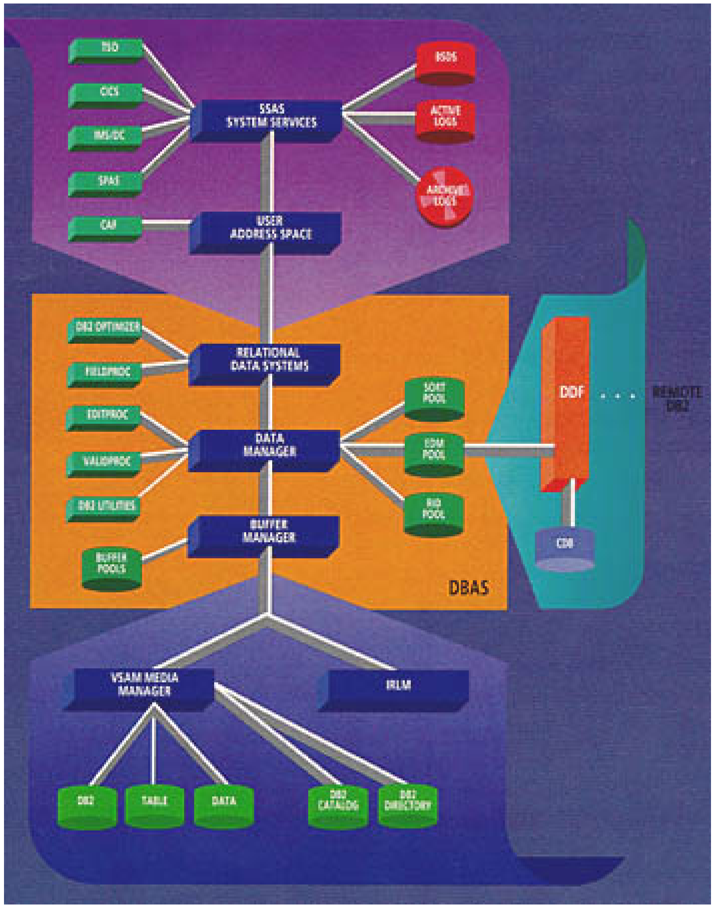
Two other components of DB2's architecture must be examined: the Boot Strap Data Set (BSDS) and DB2 logging.

The BSDS is a VSAM KSDS data set utilized by DB2 to control and administer the DB2 log data sets. It is an integral component of DB2, controlling the log data sets and managing an inventory of those logs. The BSDS is also used to record the image copy backups taken for the SYSIBM.SYSCOPY DB2 Catalog table. Because SYSIBM.SYSCOPY records all other DB2 image copies, another location must be used to record image copies of the SYSIBM.SYSCOPY table.

DB2 logs every modification made to every piece of DB2 data. Log records are written for every INSERT, UPDATE, and DELETE SQL statement that is successfully executed and committed. DB2 logs each updated row from the first byte updated to the end of the row. These log records are written to the active logs. There are usually two active log data sets to safeguard against physical DASD errors. The active logs must reside on DASD (they cannot reside on tape). The active log data sets are managed by DB2 using the BSDS.

As the active logs are filled, a process called *log offloading* is invoked by DB2 to move the log information offline to archive log data sets. This process reduces the chance that the active logs will fill up during DB2 processing; this would stifle the DB2 environment. Archive logs are accessible by DB2 to evoke table space recovery. The BSDS manages and administers the archive logs.

Figure 2 offers a complete picture of DB2 "under the covers." It contains all the components of DB2 that operate together to achieve an effective and useful relational database management system.



DB2 "under the covers."

©2005, 1995 Mullins Consulting, Inc. All rights reserved.
Home.   Phone: 281-494-6153   Fax: 281-491-0637