Craig S Mullins
*Platinum Education, 1815 S Meyers Road, Oakbrook Terrace, IL 60181, USA*

Craig Mullins looks at some of the less obvious traps and pitfalls that may be encountered when using multiple DB2s in a client/server environment.

# Implementing DB2 in a client/server environment

DB2 is no longer just a mainframe product. Versions of DB2 exist for a large array of platforms, of which MVS is only one (see Figure 1). Many shops implement applications on several different platforms and inter-connect them using client/server development methods.

| Platform | Operating system | DB2 product |
|---|---|---|
| AS/400 | OS/400 | DB2/400 |
| Mainframe | MVS | DB2 |
| Mid-range – IBM | VM | DB2/VM |
| Mid-range – IBM | VSE | DB2/VSE |
| Personal Computer | OS/2 | DB2/2 |
| Mid-range – RS/6000 | AIX | DB2/6000 |
| Mid-range – HP Series | HP/UX | DB2 for HP-UX |
| Mid-range – Sun | Solaris | DB2 for Solaris |

*Figure 1: Current DB2 products*

However, these products are not 'plug and play' commodities simply because all of them share the name 'DB2'. There are some major differences among these products in their current releases. The biggest differences are relatively easy to detect. These include:

- Differences imposed because of operating system constraints (OS/2 versus MVS versus AIX versus OS/400 etc).
- Back-level compatibility issues such as ensuring that DB2/2 will work with code developed for OS/2 EE Database Manager, or DB2/VM will work with code developed for SQL/DS.

- Workstation orientation differences such as GUI interfaces and drag-and-drop menus.
- Subsystem-centric implementation (MVS) versus database-centric implementation (workstation).

However, there are some 'gotchas' lurking under the covers that may be more difficult to find. This article will discuss some of the more esoteric traps and pitfalls that may be encountered when using multiple DB2s in a client/server environment.

## Some major differences

Of the basic differences mentioned above, the only one that may not be overtly obvious is the focus of the DBMS implementation. DB2/2 and DB2/6000 are database-centric implementations. This implies that each new database carries with it its own system catalog. Additionally, it is not possible to simply access tables across different databases. A distributed connection is required.

On MVS, DB2 is subsystem-centric. There is a single system catalog that spans databases. Each subsystem has a unique identification and multiple databases can be created within it.

## Security

Another challenging aspect of moving to a multiple DBMS environment is the implementation of a systematic and consistent security architecture. This can be particularly difficult because each of the DB2 products provide different techniques for authorizing data access.

Perhaps the biggest hurdle to overcome is the lack of secondary 'authid' support in the non-MVS DB2 products. Secondary authorization lists are very popular in DB2 MVS shops because they minimize the administrative burden of security management. The lack of secondary authids will cause different security strategies to be implemented for different platforms.

A similar problem is exemplified by the User Profile Management (UPM) feature of DB2/2. Every user must provide a valid user-id to DB2/2 before accessing data. This feature does not exist on the other platforms (including DB2/6000). Requiring UPM on one platform with no support for it on other platforms further complicates data security.

Relational databases provide the basic level of authorization through the usage of data control language GRANT and REVOKE statements. However, this can cause yet another type of security implementation problem. Each DB2 product supports the basic GRANT and REVOKE in a different manner. Consider, for example, the WITH GRANT OPTION clause. The workstation DB2 products do not support this clause – instead there is a CONTROL authority that provides similar capabilities.

Finally, the default security that comes with each product differs. DB2 MVS requires that, prior to allowing data access to any resource, a user must first be granted authority on that resource. However, in the workstation DB2 products, all users have SELECT authority on the system catalog tables by default. This would be unthinkable in a mainframe environment.

## Basic incompatibilities

The platform on which the DBMS is operating has more of an impact than some people would have us believe. Even if the syntax of all the DB2 products were exactly the same (they are not), some basic incompatibilities would arise.

Consider, for example, floating point arithmetic support. The format used by each of the DB2 products differs. The venerable DB2 for MVS uses IBM's proprietary 370 floating point format. DB2/2 and DB2/6000 use IEEE floating point format. This poses a problem that is larger than one might first expect for applications that must be ported from one environment to another. This will cause calculation results to differ from environment to environment. So, for floating point numbers, DB2 for MVS will provide a different result for A + B than will DB2/2 or DB2/6000. Whether this is a big problem or not will depend upon the accuracy required by your applications that use floating point numbers.

A similar type of problem can occur when collating sequence is important. DB2 for MVS uses the EBCDIC collating sequence. The other DB2 products adhere to the ASCII collating sequence. As most programmers know, EBCDIC sorts letters before numbers, whereas ASCII does just the opposite. If sort order is important, keep this basic incompatibility in mind. DB2/6000 can optionally maintain EBCDIC collating sequence if it is explicitly requested. DB2/2 cannot (as of Version 1).

## Programming problems

Additional multi-DBMS implementation problems will exhibit themselves at the programming level. There are myriad small incompatibilities among the different DB2s. Generally, these problems revolve around SQL limitations. For example, DB2 MVS supports up to 750 columns for a SELECT or INSERT statement. The workstation products currently limit the number of columns to 255. Another example is the number of predicates that can be coded on a single SQL statement – DB2 MVS supports up to 750 predicates, whereas the workstation DB2s support only 300. These are but two small examples of the many types of programming incompatibility.

At another level, the different DB2 products can display differing support for embedding data access into a host language. DB2 MVS does not support an API level interface, whereas workstation DB2 does. On the other hand, the workstation DB2 products support a form of stored procedures known as the Database Application Remote Interface (DARI). DB2 MVS will not provide stored procedure support until the next version is available.

In terms of host language programming, substantially larger applications can be developed for DB2 MVS. This is probably not surprising because MVS operates in a larger scale environment. However, the DB2 products seem to be engineered with this difference in mind. Consider, for example, that the number of host variables per precompiled program supported by DB2 MVS is constrained only by available storage. The limit for workstation DB2 precompiled programs is 880.

There are also pure incompatibilities in terms of the SQL supported. For example, the workstation DB2 products provide the EXCEPT [ALL] (set difference) and INTERSECT [ALL] (set intersection) clauses, whereas DB2 MVS does not. On the other hand, DB2 MVS supports special registers that the workstation DB2 products do not: CURRENT DEGREE, CURRENT PACKAGESET, CURRENT SQLID, and CURRENT TIMEZONE.

The final type of programming problem that must be contended with is environmental. These types of problem lie in the very nature of the construction of the DBMS.

The best example of an environmental inconsistency is the manner in which timeouts and deadlocks are handled. When a locked resource cannot be acquired by a requesting task, DB2 MVS will wait for a predetermined amount of time, and then cause the request to terminate or timeout. This feature is designed into most DB2 MVS application programs. A program may test for the timeout return code and try again. DB2/2 does not timeout – it will keep on waiting and waiting.

Deadlocks occur when task 1 has resources locked that task 2 requires, and task 2 has resources locked that task 1 requires – see Figure 1. When a deadlock is detected by DB2 MVS, the task with the least amount of CPU time is killed. The deadlock detector in the workstation DB2 products will randomly kill one task (sometimes, the task with the most CPU time). This can cause headaches.

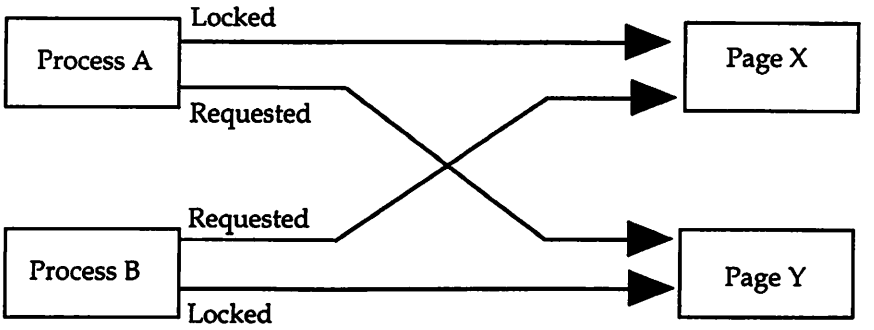


*Figure 2: Deadlock definition*

## Administration and utilities

The final implementation consideration is an administrative one. Databases must be administered, and anyone with any level of DBA experience that spans more than one DBMS knows that no two DBMSs are exactly alike. This is true with the various DB2s.

Although all relational databases share similar characteristics, the exact implementation differs. As of Version 1, there are no table spaces on the workstation products. For Version 2, there will be a table space object, but it will not be the same as the venerable DB2 MVS table space. And DB2 for VSE and VM uses something similar to a table space known as a data space.

These differences are confusing enough, but it is even more difficult than it seems. Consider:

- There are different types of table space (segmented, simple, and partitioned), each of which must be managed differently. Furthermore, DB2 MVS uses index spaces, but DB2 VSE (for example) puts both data and index information in the same data space.
- There are different clustering methods. The workstation DB2 products only cluster during REORG. DB2 MVS associates a clustering index with a table and will attempt to maintain clustering during data modification (INSERT, UPDATE, DELETE).
- There is no support for defining 32KB page sets in the non-MVS DB2 products.
- DB2/2 and DB2/6000 automatically generate the unique index when a primary key is created. This is not true for the other DB2 family product offerings.
- DB2 MVS uses one dataset per index unless it is a partitioning index, in which case multiple datasets are utilized. DB2/2 uses one file for all indexes on a given table.

And that is only a sampling of the differences. As you can see, there are many – some more serious than others.

From a utility perspective there are differences as well. Consult Figure 3 for a rundown of the differences between the DB2 MVS utilities and the DB2/2 utilities. The basic purpose of the utility is listed along the left hand side and the appropriate utility for each is listed in the right hand columns. Notice that DB2/2 has no REPAIR SET, CHECK, or LOAD ENFORCE NO utility support. This is because referential constraints are always enforced in DB2/2. They can be bypassed by DB2 MVS utilities.

| Purpose | DB2 MVS | DB2/2 |
|---|---|---|
| Integrity | RECOVER | ROLLFORWARD / RESTORE DATABASE |
| Statistics | RUNSTATS | RUN STATISTICS |
| Reorganization | REORG | REORGANIZE TABLE |
| Back-up | COPY | BACKUP DATABASE |
| Unload | DSNTIAUL (REORG [UNLOAD ONLY]) | EXPORT |
| Load | LOAD | IMPORT |
| Zap/Fix | REPAIR | TARR |

*Figure 3: Utility comparison matrix*

## Synopsis

The bottom line is quite simply that IBM's relational database management system offerings are not yet a commodity. This is the direction in which IBM is moving, but it is definitely not there yet. There is a learning curve as developers move from DB2 to DB2. It is true that the learning curve is not as steep as, say, moving from DB2 MVS to Sybase SQL Server, but the curve is there none the less.

To mitigate the potential for the unexpected, it is a wise course of action to understand that DB2 is not always DB2. The name may be the same, but, as we have outlined above, the underlying software can differ significantly. Plan to expect the unexpected by:

- Experimenting with a new DB2 offering prior to development.
- Embrace training opportunities to learn each new product prior to implementation and during the application development life-cycle.
- Keep informed about new product releases from IBM that will remove some of the differences (and possibly create more!).

By being prepared, you will ensure the success of your DB2 client/server initiative! ■