

Consolidating Sybase

When designing system platforms for Sybase based applications, three patterns are available.

1. Traditionally, a single Sybase instance is installed onto a system host. This serves an application and user community. This remains a popular pattern for ISV products, however, where a rigorous environment management policy is adopted, this leads rapidly to 'server sprawl' as each DBMS instance requires not only production but also a development, testing and contingency system.
2. Alternatively, multiple applications can be collected into a single Sybase instance. This is referred to by Sun as 'Aggregation'. It is best done in conjunction with a single data model so that, for instance, only one logical customer (or counter party) table exists.
3. The final implementation model is to install multiple Sybase instances within a single instance of the operating system. As noted above, with Solaris 10, these can be installed within their own Zone, or they can share zones. This Sun refers to as database server 'Consolidation'.

Sybase's development of multiple workspaces has again extended the parallelisation of the server and reduces the number of serial bottlenecks. This is a technique that might permit multiple applications to be hosted within a single ASE server instance i.e. adopt pattern two above. ASE has for many years had multiple 'databases' within it but a database is a unit of recovery, not of applications logic and while Sybase has dramatically reduced the number of scarce resources within the ASE instance, it has no concept of application and its resource management algorithms have no concept of priority nor of service quality.

These weaknesses can be overcome by the Solaris resource manager which permits an application via a project to have its system resources guaranteed against anti-social behaviour by other applications. Having multiple Sybase instances within a Solaris instance would allow a more sophisticated resource management policy to be declared. Further reasons for 'consolidating' is that the regression tests for application changes are simpler. A change in one application will not require tests in the databases and procedures that are required by others and start/stop requests do not impact other applications. Any conflict around the values of the server run time parameters can be resolved using the consolidation

model, to the extent that even different versions & EBFs of Sybase can be applied to different applications (using the file system name space to enforce and differentiate between versions). This last factor is very important if the platform designers do not own the Sybase version definition policy, such as when ISV code is being used.

Platform designers have a choice and they should carefully consider which of the three patterns they wish to implement as they develop their infrastructure plans.