Replication Versus Clusters

A Comparitive Look at Each

Data Management Group

PROGRESS
SOFTWARE

Replication Versus Clusters

Where computing terms are named after standard English words, the original definition sometimes gives an interesting insight into the new concept.

The dictionary I use once belonged to my father. It is a first edition of The Collins Dictionary of The English Language, published in 1979. What this means is that very few computing terms are listed and IT-specific definitions for common words are also not included either.

In my dictionary there are five definitions for the word *replication*. Only two of these seem pertinent to the word's IT-specific use. The first is: "the repetition of a procedure in order to reduce errors," and the second is biological in context: "the production of exact copies of complex molecules that occurs during growth of living tissue." Combining these gives us a decent idea of what replication means to someone working in IT – to repeat the writing of data, thus producing an exact copy of your data, in order to reduce errors (and downtime) as your database grows and matures.

Cluster is a different matter. There are eight definitions in The Collins Dictionary of the English Language (second edition). All of them have the same underlying theme (which gives only a basic indication of what clustering means to IT): the collecting of similar things into groups.

We have the English-based etymology of these IT terms, but what does replication and clustering mean in terms of databases? What are the relative merits and demerits of each approach to high availability systems?

Failover Clusters

Clusters provide the ability to eliminate a single machine as a failure point, protecting your database access from server hardware failure.

A system cluster is made up of two or more machines (known as nodes), which are integrated through hardware and software to function as a single, virtual machine. In a cluster, redundant hardware and software are put in place to enable failover: if something goes wrong on one node, or the node needs to be taken offline for maintenance, cluster resources failover to another node to provide continual access.

Progress originally offered Fathom® High Availability Clusters to enable the Progress® OpenEdge® databases to be cluster-able. This functionality has been blended into the OpenEdge 10B enterprise database product, and is now referred to simply as failover clusters.

Failover clusters functionality is operating system and hardware vendor-independent. The failover clusters interface is easy to use, greatly simplifying the administration of OpenEdge in a clustered environment.

Failover clusters do not replace OS-specific cluster management software, it simply integrates OpenEdge into your chosen cluster manager software, so that OpenEdge is properly defined as a cluster resource. In this way, cluster resource administration and failover mechanisms are enabled for your database, which will automatically failover during planned or unplanned outages.

Failover clusters mean that cluster-enabling an OpenEdge 10 database does not require Progress database administrators to become cluster experts, though they do need to have a base of knowledge regarding the cluster manager. It is important to understand, however, that cluster implementations are complicated, and someone in the organization must have experience in how to implement and manage them. With failover clusters, database administrators simply decide ahead of time how clustered resources will behave during failover by setting agreed failover policies, which predefine the recovery action that the resource and all of its dependencies should automatically follow when a failover condition occurs. Such failover policies include, for example, the amount of time to attempt to restart a database on its current node, and whether or not the application should be transferred back to the primary node upon its return to operation.

Failover clusters has a simple command-line interface (PROCLUSTER) to your operating system's clustering software. PROCLUSTER registers resources that need to be persisted in a cluster environment. You must register your database as a resource, however you do not need to identify the dependencies for the database – failover clusters takes care of this on your behalf.

Failover clusters integrates OpenEdge into the cluster by making use of the pre-existing cluster manager software, and by augmenting OpenEdge feature functionality. When you use clusters, you can still use the PROSERVE or PROSHUT commands and their equivalents; you can also use PROCLUSTER to start and stop the database. You can also use the admin server to start and stop the database. The operating system cluster manager software knows about OpenEdge, and will handle it properly in the event of a fail over.

If the database structure changes or the database is moved to a new common storage device, PROCLUSTER makes it very easy to ensure that the database is still protected if a failover occurs. For example, should the

cluster-enabled database structure change (using PROSTRCT), failover clusters will automatically identify that the resource needs enabling, take the resource offline if it is running, and disable and then re-enable the resource automatically.

OpenEdge conforms to the security model defined by the OS vendor in terms of what users can create and modify; access rights to the various directories and devices; and rights to start and stop resources, such as databases.

Performance of the database should not be affected by the use of clusters beyond the additional separate process required to probe and report on the database's viability to the cluster management software.

For failover clusters, OpenEdge requires at least a two-host configuration, utilizing common storage architecture such as SCSI, and systems with redundant network and cluster interconnections. Your database must reside on one or more shared devices (also known as common storage).

Failover clusters' most significant advantage is its ability to offer ease-of-use and maintenance by automating many of the manual procedures you would otherwise need to perform to ensure proper failover. However, it should be borne in mind that clustering does not protect you from physical failure of the disks where the database resides.

Replication

Data replication has two major attractions: to distribute copies of information to one or more sites, and/or to provide failure recovery to keep data constantly available to customers. OpenEdge Replication automatically replicates a local OpenEdge™ database to remote OpenEdge databases running on one or more machines. OpenEdge Replication also removes the database as the single point of failure in your high availability environment.

Thus, OpenEdge Replication offers users the ability to keep OpenEdge databases identical while also providing a hot standby in case a database fails. When a database fails, another becomes active. Therefore, mission-critical data is always available to your users.

OpenEdge Replication disallows updates to the target database by anything other than Replication. However, with OpenEdge Replication Plus, the target database permits queries and reports by users, as well as any non-database write activity (database utilities, for example). This setup can greatly improve performance by allowing transaction-based applications and read-intensive activities (such as batch reporting) to be performed against separate yet identical databases.

Of course, the source and target databases can be on the same machine, but this would be relatively pointless, as one of the great advantages of replication is that the target database will kick in if the source machine fails.

OpenEdge Replication supports two methods of replication: synchronous and asynchronous. During asynchronous operation, OpenEdge Replication sends AI blocks from the AI transaction log to be applied to the target database. During synchronous operation, the OpenEdge Replication blocks further processing for the user until the transaction is fully applied to the target database.

Of the two configurations (synchronous and asynchronous), asynchronous performs better. Synchronous is the safest, but, because the user blocks in the synchronous model, performance is slower than in the asynchronous model.

Before you begin consider implementing OpenEdge Replication, you should be aware of a few limitations. In general, a target database enabled for OpenEdge Replication cannot be modified in structure or data when OpenEdge Replication is not running, and OpenEdge Replication does not support 2 Phase Commit enabled databases. A source or production database can be opened as single user as long as after imaging and replication are not disabled. Schema changes are also allowed on the source database and get transferred to the target database automatically.

You should also ensure that you have enough resources to implement AI processing on the source database. When you turn on AI, the transaction logs generated could consume significant disk space. The machine you choose should also have enough CPU and memory to support the addition of OpenEdge Replication.

Also, reliable TCP/IP communications between the source and target database is a key element in keeping your source and target databases up to date. Without reliable communications, OpenEdge Replication will spend time in failure recovery, which will cause interrupted user access to your databases.

Given the right circumstances, however, Replication is a very attractive option. On top of the availability of mission-critical data 24 hours a day, seven days a week, and the near eradication of disruption in the event of unplanned downtime or disaster, it also provides on-line backup and deferred system startup for greater system availability and enhanced capability for AI extents for improved data validation.

Summary

In the true high availability / business continuity world you should have both of them. Clusters are designed to prevent machine failure, replication is for database failure.

Clustering and replication are both very attractive options, and both can provide a significant step towards achieving the ultimate goal of 24/7 data availability. In terms of cost, replication tends to be somewhat expensive, as clustering is included as part of OpenEdge 10 enterprise licenses, whereas OpenEdge Replication is sold as an additional product.

In the ideal "high availability" world, you would, of course, implement both measures. They are essentially two sides of the same business continuity coin. Clustering minimizes downtime through machine failure, and replication minimizes downtime through database failure.

Corporate and North American Headquarters
Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA Tel: 781 280 4000 Fax: 781 280 4095

Europe/Middle East/Africa Headquarters
Progress Software Europe B.V. Schorpioenstraat 67 3067 GG Rotterdam, The Netherlands Tel: 31 10 286 5700 Fax: 31 10 286 5777

Latin American Headquarters
Progress Software Corporation, 2255 Glades Road, One Boca Place, Suite 300 E, Boca Raton, FL 33431 USA Tel: 561 998 2244 Fax: 561 998 1573

Asia/Pacific Headquarters
Progress Software Pty. Ltd., Level 2, 25 Ryde Road, Pymble, NSW 2073, Australia Tel: 61 2 9496 8439 Fax: 61 2 9498 7498

PROGRESS SOFTWARE

www.progress.com