Implementing Business Continuity on a Shoestring

Stephen Swart
Systems Administrator
Business Applications Department
Business Affairs Division
University of North Carolina at Wilmington

Abstract

The University of North Carolina at Wilmington is in the eye of many storms or so it would seem. Since 1996, Bertha, Fran, Bonnie, Dennis, Floyd, Kyle, and Charley have passed within 60 nautical miles and the main campus of UNCW which lies less than seven miles from the coast. In this environment you really don't want all your datacenter eggs in one basket. The Business Affairs Division has worked proactively to provide warm backup site for their systems, within a budget using off the shelf technology, to provide better data redundancy and system continuity.

Introduction of the Organization

The University of North Carolina at Wilmington (UNCW) is a comprehensive public institution with more than 1,600 faculty and staff and nearly 11,500 students located in southeastern North Carolina. With 71 undergraduate degree programs and 26 graduate programs, UNCW has a variety of academic programs designed to meet the diverse needs, abilities and interests of all students. The university also offers one doctoral program in marine biology through the Center for Marine Science, one of the newest and most technologically advanced coastal ocean science research facilities on the eastern seaboard. The Business Affairs division of UNCW is the support arm of the university and supports the university mission by providing excellent facilities, financial and business services to a variety of constituents--faculty, staff, students, alumni, parents and visitors.

Initiative

The University of North Carolina at Wilmington is located in a region that is frequently threatened by hurricanes; therefore, we are constantly working to improve the safety of our data and the recoverability of our systems. The Business Applications department maintain the servers, which support not only the operations of the facilities, parking, project design, accounting, purchasing credit cards, travel, debit card and printing, but also web applications, which integrate to these systems. Operational hours are extending, customer base expanding, and web applications are available round the clock; therefore, it became imperative to provide better data redundancy and recoverability for these systems.

We had implemented new software and hardware to improve our ability to back up our data to tape in the recent past, but building systems and recovering from tape is a time consuming process. We needed a system in place, within our budgetary constraints, where our most important production systems, with the data as up-to-date as possible, could be functional in a minimal amount of time in case one or all of our primary systems were no longer functional.

Design

The most important factors in building this system were place and space. We needed a place, somewhere geographically separate, affordable, with limited physical access, sufficient cooling, power, and high speed network connectivity to locate some servers as well as needing affordable, sufficient, and network available disk space to store all the data and applications. We also needed processing power and the ability to effectively have a system masquerade as another on the network. Finally, we needed the ability to copy data and to schedule when the copies would occur. This all had to occur, within our yearly server budget of $25K.

Implementation

The first, a maybe most important, hurdle was resolved by getting use of some free space in the telecommunications closet in the main administration building. The building is located approximately one mile from our primary site. It had sufficient cooling and ample network access. As a bonus, this location was in a section of the building that also had generator backup power. Additionally, the access to the area was restricted to telecommunications personnel. As for disk space, we were able to purchase .8 terabytes of disk space by purchasing a SCSI RAID array of IDE disk drives for around five thousand dollars. Now we had a location and space for data and programs. Next we needed some computing horsepower. We then were able to procure an old rack that was left over after a system upgrade and purchase a Dell Poweredge 2650 server to serve as the backbone of our recovery system. We installed Microsoft Windows 2003 and Microsoft SQL 2000 server software using our Microsoft Campus agreement and attached the server to the drive array. We also loaded a copy of Oracle 9i to match that, which was required for our parking system, but disabled the Oracle services to comply with our license agreement. At this point, we had created the structure for our warm recovery site strategy. To add to the sites availability we have moved away from using the Windows Internet Naming System (WINS) in favor of using DNS names whenever possible. This improved reliability of naming services to the applications that required them and also allowed us to move from server to server simply by changing the servers IP address. Additionally, the network staff provided us with a Virtual LAN or VLAN so that, even though the primary and recovery servers were located at different sites and on different switches, they would exist in a virtual same subnet. Therefore, the recovery server could be reset to use the same IP addresses as one or more of the primary servers. At this point, we began fleshing out our recovery system. We were already doing scheduled Microsoft and Oracle “cold” database backups of all our databases on a nightly basis. We simply put a share on the recovery server and had scheduled a nightly batch job, using the windows scheduler on each of those servers, to copy those files to specific folders on that network share. Simply doing this, would allow us to restore any of the databases from the time of the last backup on the recovery server without having to find, load and restore tape. Also, since the server already had an operating system and the database software loaded recovery time would be reduced significantly. This was a good first step, but we could do better.

Next, we looked at how we could make the database restore automatically for all the production databases. After doing a little research, we were able to develop a SQL script for our Microsoft databases, which would read a table of databases to be restored along with the file/folder locations of the backup files. The script would then create the databases if they did not previously exist, read the appropriate backup file to get the logical database file and log file(s), and then dynamically generate and execute the SQL command to restore the data. Now we had a system in place, by which all our production databases were operational, as of the previous night's backup on our recovery server. By simply changing the IP address of the Ethernet interface card the recovery server would respond as the primary server without having to touch applications or the users' desktop systems. Also, because the recovery system is Windows 2003, changing IP addresses do not require a system reboot. By using this method, with multiple Ethernet interfaces and ports, the recovery server would serve and appear as multiple SQL servers. Although this may cause slowness in peak demand situations, the service would still be available nearly immediately after a disaster and databases could be moved to other servers as more resources became available. Additionally, as we upgraded system hardware we have moved still viable servers to our recovery location to provide additional computing horsepower without additional cost.

The parking system was a little bit different. It uses an Oracle 9i database engine for its data storage and retrieval. Working closely with the vendor and using their backup procedures we were able to build a copy of the Parking systems' Oracle 9i instance on the recovery server. We then made Oracle services on the recovery server inactive requiring a manual startup. On a nightly basis, the operational files from the production parking server cold backup are copied to the appropriate place in the folder structure of Oracle server on the recovery server. By simply starting the Oracle services from the windows services console and changing the IP address on one of the network interface cards, our recovery server can replace the production parking server.

In addition to housing live copies of the previous night's databases, the recovery server also serves as a repository of nightly backups to disk of or our web, application and ftp production servers as well as our CAD file network repository. These can be restored quickly to alternate systems, if needed, without accessing tape.

At the recovery site, we also strategically placed a single external tape drive to match the tape format and type we use in our multi-tape loader. In a situation in which our primary computing facility was destroyed, this would allow us immediate access to our weekly archived tapes, which are also stored off-site.

Because this system was loosely designed and not constrained by one vendor's solution for disaster recovery, it is still evolving. Plans include moving all the production databases to Microsoft SQL 2005, which would allow us to move to real-time database synchronization using the new database mirroring feature replicating the primary and recovery servers, and allow database clients to automatically switch to the alternate recovery server, should the primary database server fail. Because another Oracle license is cost prohibitive, we are considering partnering with our Information Technology Systems Division to provide a hot-site repository for the parking systems data on their new Sun based Oracle servers. We also plan to use software to replicate our web services and are experimenting with using Windows 2003 network load balancing features to allow load balancing and failover with a secondary server to be located at our remote recovery site. Hopefully, within eighteen months, our recovery site will go from warm to hot.

Benefits

Because of this relatively simple system, we now have in place the ability to, within a few minutes without a server reboot, have full operational access to data from the previous night's backup. This is important in our area because of the possibility of hurricanes, but also protects us from many other types of disaster. The recovery site has also come in handy during some maintenance operations. For example, a firmware issue in a RAID controller, for the server that houses the database for our facilities management software, caused an issue with the drive array that required the array be rebuilt. A rebuild that took about 5 hours caused less than an hour of downtime, which we were able to schedule at time of low usage. All the things we did were relatively simple and inexpensive due to our use of off-the-shelf tools. Fortunately, our department did not have to rent space or provide network equipment, which would have driven up cost. With a single tape drive, UPS, campus software licenses, disk array, and computer, our expenditures were less than $14,000.