Moving to Client/Server Computing, A User's Perspective

Office of the Registrar
University of California, Davis

Bruce Purcell
Director of Computing Services
Franklin College, Switzerland
bpurcell@fc.edu

The move to client/server computing in recent years has boomed. Registrar's Offices, other university departments and Information Technology (IT) departments have stretched their resources to the limits in purchasing the required hardware, software and performing the conversions.

A successful conversion to a client/server environment requires a great deal of expertise, not only from the IT team, but from the client team as well. The challenges faced by IT departments are well documented, and much has been written on managing the installation and conversion. Less well documented is the work that must be performed by the users of the new system.

The users are the experts in the functional aspects of the new system. Only the users know what data is required and what results are finally expected, including exceptions. It is incumbent on the users to test every new program completely and report problems to the IT department so that these problems can be corrected. It is not enough for a department to say simply "it doesn't work" or "this system is no good".

Key to a successful implementation is user participation and specific communication with their IT department. This includes:

- Testing of all programs implemented.
- If on a local area network (LAN), testing of LAN servers.
- Stress testing of client computers.
- Stress testing of any central fileservers installed.

This session will cover the problems encountered by the Office of the Registrar in converting to a client/server student information system and the solutions found by both Office of the Registrar staff and Information Technology staff.

Moving to Client/Server Computing, A User's Perspective

Office of the Registrar
University of California, Davis

Introduction and Background

The University of California, Davis is part of the nine campus University of California (UC) system. The UC system is governed by a Board of Regents.

UC Davis is a leading national research university. It is located in Davis, California, a small community about 20 miles west of Sacramento, the state capital. There are approximately 24,000 full-time students. UC Davis has a full range of programs and degrees – baccalaureate, doctorate and professional.

The Student Information System (SIS) is SCT's Banner product. Banner is built on an Oracle database. Banner has been heavily modified by both the IT department and the Office of the Registrar programming staff with estimates that as much as 34% of the Banner product has been modified. Of the 1,792 forms and modules, approximately 600 have been modified. Many of the other 1,100 forms and modules are not used by UC Davis. Additionally, the Banner financial aid and billing modules are installed.

The hardware is a Sequent "refrigerator size" server with 12 Pentium processors. There are approximately 800 total users of Banner with an average of 200 concurrent users. Prior to the conversion to client/server, access to the database was primarily through telnet sessions and Oracle forms. There was some access to departments for downloading of data using SQL and ODBC. SQL scripts are heavily used by the programming staff.

The Office of the Registrar's primary responsibility is the registering of students for classes and the maintenance of the students' academic histories. Additional responsibilities for the UC Davis Office of the Registrar include the issuance of transcripts and grades and classroom scheduling. Additionally, the Office of the Registrar has its own IT staff that provide programming support, maintain the telephone registration system, maintain the departmental LAN and provide some support and training services to campus departments.

The conversion to client/server is a project that was managed primarily by the campus IT department. The role of the Office of the Registrar was to test the new forms as they were converted and test the overall implementation. The office took on the additional responsibility of configuring client computers, assisting in the management of installing the client software in other departments involved in the initial testing and making recommendations regarding central services such as proxy servers.

This presentation will cover the work performed by the Office of the Registrar in the overall implementation.

Description of Registrar Office Environment and Technical Services

Our office shared a single file server with the Admissions Office. There is a combined total of approximately 80 users on a Windows NT LAN. The clients are a combination of Windows for Workgroups 3.11, Windows NT 3.51 and Windows NT 4.0 with plans to move all users to Windows NT 4.0. Each workstation is a minimum Pentium 100 with 32 MB of RAM.

The Office of the Registrar technical staff totals four people. There are two programmer/analysts that manage all projects and provide programming support for the SIS and manage the telephone registration system. Additionally, there is one systems administrator that manages the NT server and all clients and provides user support and one trainer that provides training to campus departments in the use of SIS.

The Admissions Office has one support person and one programmer.

Initial Recommended Client Configuration from IT

The initial client configuration from the central IT department recommended loading all forms (over 300) onto each client computer in addition to all executable programs for maximum performance. There was no automated distribution of updated forms or executables. Remote (dial-in) users were not being supported.

User Modifications

Our response was that it would be virtually impossible to manage eighty users with all executables and forms loaded onto each individual computer. The distribution of forms alone would be impossible to manage. Additionally, we required dial-in service for our users. We have several users that work extensively from home or on the road.

We moved all forms and executables onto our Windows NT server into a read-only area. Since we had a mixture of Windows 3.11 clients and Windows NT clients, we required that both be able to access the forms and configuration files. Initially, only 16-bit forms were available, so this was no problem. We had refused to install Windows95 clients anywhere on our network. Since Windows95 was proving to be unstable, this turned out to be one less headache for us.

The Windows 3.11 users all used a centrally located oracle.ini file. This way, the only installation required at a workstation was putting up an icon and referencing the oracle.ini location in the win.ini file. Updated forms, executables or configuration changes were done one time on the server.

The NT users presented a bit more of a challenge since there is no oracle.ini file used. Instead, all configuration settings are stored in the registry. We set up one NT machine with all required settings, then exported the registry key to a text file. With administrator's privileges on the network, we were able to access all NT client's registry files from one workstation and import the registry settings. It is a bit more time consuming, but the installation is still managed from one location as opposed to physically visiting each computer.

Access times were excellent, with all forms loading in an average time of two to three seconds.

Next, we installed a dial-up server. We used an old 486 server with NT 4.0 to manage eight 33.6 baud modems and used Microsoft's Remote Access Server (RAS). In addition to managing the dial-in service, we put our office web site of over 800 pages on this server.

We installed NT on all home clients. We wanted to use Microsoft's TCP/IP stack and winsock services consistently through our network. Since Windows 3.11 RAS does not support TCP/IP, we would have had to use Trumpet for these services. It seemed easier to just go ahead and install NT. We loaded the Oracle executables on the local computer, but used the forms from our server. The initial log on process was slow since it used a 700k form, but response time after log on was reasonable. Since we had the 486 service already, this solution cost us under $2,000 to implement.

We looked at using Ntrigue for dial-in services, but it simply wasn't cost-effective for a small office with six to eight dial-in users.

It is important to note that during this time, we also had users constantly testing the forms to ensure that they worked properly and that access times were acceptable. Our Associate Registrar worked directly with users to test all forms and reported all bugs to IT for correction.

Additionally, we reported all of my client configuration changes to the central IT department along with my recommendations. The IT department was doing additional work in client configuration, but we never used their recommended solutions because our configurations worked better and were more manageable. IT client configuration always recommended loading all executables and configuration files on the local computer rather than on local fileservers.

IT Configuration Changes

At this point, the IT department decided on the use of proxy servers for central storage of forms and the distribution of executables. UC Davis was starting to implement Oracle-based programs other than the SIS, making the concept of central servers critical. They decided on Linux-based servers using the Andrews File System (AFS) and programs written in-house to distribute the executables from these servers. These servers would automatically receive any changes in forms from a central repository. Users could map a drive to these servers with no additional drivers and use the forms directly. Their plan called for one proxy server for every eighty users.

The only problem we saw with this configuration initially was that the recommendation was still to load executables onto each workstation and their automated programs did not have any option for server-based installations. We started using the forms from the proxy servers, but still insisted on keeping our executables and configuration files loaded on our server.

Problems with the Proxies

The performance from the proxy server proved to be incredibly slow. Forms were taking over one minute and up to five minutes to load. This was with no load on the server. IT sent out a network engineer to help identify why performance was so poor. He came out and put a sniffer on our network. we also started asking about the proxy server hardware. Here's what we found out:

1. The searches for files were case-sensitive. This was primarily a problem when a program within a form called a subroutine from a .pll or .plx file. We moved these files to the same directory as the Oracle .dll files.

2. The recommended oracle.ini file referenced incorrect directories for file searching, thus depending on Oracle defaults. This was only a problem with Unix. We changed the oracle.ini file to reference the directories correctly.

3. The server was on 10Base2 (thinnet) wiring. This probably wasn't a large factor since there were only two devices on this segment, but it is bad practice to locate a server on 10Base2 since this has the potential to affect performance.

4. The servers had IDE hard drives. This was a huge error. IDE drives will not maintain performance with multiple users.

By implementing my configuration changes noted in items one and two, the loading time for a form was reduced from one minute or longer to three to five seconds. We were ready to stress test the servers!! We were fairly confident that the proxy servers would not hold up under stress because of the IDE drives.

The stress tests gave exactly the results we expected. With a load on the server, forms were taking over one minute to load and sometimes up to ten minutes. After further testing by IT, they determined that the IDE drives were the primary problem and made plans to upgrade the servers. We ran our own stress tests using

the forms and executables from our server without any problems. Running all forms and executables from our server, we were seeing forms load in two to three seconds consistently.

Conclusion

At this point, I left the university for Switzerland. I left feeling fairly confident that the proposed changes would give good performance.

During all of these changes, IT was also configuring servers using Windows NT and Ntrigue for Macintosh users. They made the decision to replace all proxy servers with Ntrigue servers and this is what was finally implemented. Ntrigue is a solid solution for dial-in users and Macintosh clients, but is costly to implement for general use. The cost to users was about $500 per workstation. I believe that it would have been more cost-effective to use the proxy server solution.