DO-IT-YOURSELF CLOUD COMPUTING

Dr. Paul Dorsey, Dulcian, Inc.

Cloud computing is one of the next big "buzzwords" in the IT industry. This is not a new idea. Anyone using Gmail, Yahoo mail, Hotmail, etc. has already been exposed to "cloud computing." The centralization vs. de-centralization debate has been raging since the early days of computing. We continue to go back and forth between centralizing everything to localizing data at the division/department/office and even desktop levels. Now that reasonable internet bandwidth exists in many (but not all) parts of the world, it is feasible to start placing more and more computing resources in centralized locations. This paper discusses some of our experiences at Dulcian in becoming a "cloud provider" and attempting to utilize this concept for both internal projects and client work.

"Cloud Computing" is a Concept

The first thing to keep in mind is that "cloud computing" is a concept and not a technology. It does not necessarily entail setting up VMware with each application residing in its own unique virtual machine, resulting in hundreds of Oracle instances or application servers.

It is possible to set up a cloud simply with adequate security and good performance without excessive hardware and software expenditures.

Technical Challenges

There are a number of technical challenges associated with cloud computing, namely placing huge machines with terabytes of storage, hundreds of gigabytes of main memory, hundreds of cores, and more processing than we effectively know what to do with.

In addition, when centralizing applications from disparate user populations (often from different organizations) as is the case with large cloud providers, users from one application should not impact those utilizing another application. What if one application suddenly begins using a large number of resources and the performance of other applications sharing the same cloud are impacted?

There are also numerous security concerns. Inappropriate or inadvertent data access across applications cannot be allowed occur under any circumstances, even by the most dedicated hacker.

Resolving all of these hardware, performance, and security issues associated with cloud computing comes at a high monetary and resource cost, and often requires very complex setup and monitoring. All this may be overkill for what actually needs to be accomplished in a given context.

Do-It-Yourself Cloud

This paper discusses a case study of a "cloudish" application that my company needed to support without the tremendous monetary and development resources mentioned above.

Dulcian has a user interface (UI) application development environment called Formspider®. Formspider was created using a "thick database" approach, meaning that all of the logic and UI implementation is handled within the database. The application server is nothing more than a request processor.

The Goal

The goal was to allow users to try out Formspider without downloading any files. Users were directed to a website, instructed to set up an account (with user name/password), thus making a Formspider development environment accessible to them remotely. Access would be granted to the IDE to allow users to specify their applications via the metadata using standard DBA access to the data schema. This was usually accomplished using TOAD, SQL Navigator or other similar tools.

The idea was to be able to support hundreds of users trying out Formspider without using a "traditional" cloud solution, since providing each user with his/her own environment would have been prohibitively expensive and massively resource-intensive.

The Implementation

The Formspider implementation is set up in two schemas:

1. Engine – all users shared the same engine schema.
2. Data

The engine schema includes many large PL/SQL packages. The data schema holds the application's regular Oracle database tables, metadata tables to store the application specifications, and runtime tables to support the model layer of the application. The Formspider architecture creates very little network traffic and the IDE runs easily over the internet as a web-based application.

With this architecture, users could create data tables, and packages to support Formspider scripting (all written in PL/SQL), while directly accessing the metadata repository when appropriate. Users were given the IP address of the engine in order to get into the data schema with their unique user name and password.

This solution was much simpler than setting up a full cloud environment. All users were placed into the same Oracle instance. Each time a new user entered the system, they were assigned to an already existing data schema. Since many engine objects would be needed for the data schema, a script was written to create new data schemas on demand.

Handling Wait Times

Initially, the time required to run the script, create, and populate a new data schema was approximately one minute (mainly because of the necessary grants needed). We didn't want users coming to the Formspider website to set up a development environment to have to wait that long. Therefore, the solution was to create a pool of pre-prepared schemas which could be assigned to new users at runtime. However, even with the pre-prepared schemas, when initially released, Formspider was not recommended for production usage.

Dealing with Security

For our "cloud-like" Formspider solution, security for the development environment was established by limiting each user's access to his/her own data schema and only executing privileges to the engine schema. After a year, the security features have matured a bit, but as stated above, full 24 x 7 support is not possible and the security is still not quite ready for use with production data.

Performance Concerns

At Dulcian, we work with virtual machines on all of our servers. Even this "mini-clouding" behavior has not been without challenges. When performance degradation occurs in a cloud or cloud-like environment, how do you determine where the problem lies? Is it in the database itself or a result of the cloud environment setup?

Troubleshooting performance issues in a cloud environment often requires an in-depth knowledge of Oracle and its operating system internals to conclusively identify and resolve the problem(s). When using someone else's cloud, it is not likely that you will be given access to the cloud environment internal statistics that may be needed to assess whether or not the cloud is the cause of the performance problems.

Lessons Learned

The notion of becoming a cloud provider made us appreciate how hard it is to develop a clean, secure, and efficient cloud environment. We learned quite a few things along the way.

First, it is not always necessary to have a full cloud environment in order to support "cloud thinking." An easier and less resource intensive solution may be sufficient to support your requirements.

Second, moving to a cloud environment means more than simply making the software work. There have been some recent notorious cloud failure events which resulted in irrevocable loss of data for a period of time. Even if your cloud provider is

contractually on the hook to compensate your organization for damages, a catastrophic failure on a large scale could easily overwhelm the resources to provide restitution.

Placing data into clouds managed by others can be risky. Since you do not control the environment, you will not be aware of how backups work, when they are scheduled, and how fast a backup can be restored. For very large databases, this can be a significant concern.

How secure is the cloud provider's environment? Placing large quantities of very sensitive data on the cloud might be required. Even if the cloud is protected from hackers, what prevents a knowledgeable DBA or IT professional at the cloud facility from gaining access to sensitive data and inappropriately manipulating it?

Using another example from Dulcian's experience brought to light some additional concerns. One of Dulcian's large applications built for the US Dept. of Defense will be hosted on a cloud. The impact analysis conducted as part of testing for this system was not reassuring. In order to keep backups synchronized easily and quickly, it was decided that it was not necessary to backup every table in the database every time. Since many runtime tables do not need to be backed up, they were not needed as part of the standby database because of the large number of resources and necessary bandwidth involved. The ability to do backups in the cloud environment requires some level of involvement and interaction with the cloud provider. In this case, a normal, standard Oracle standby database was not the best solution. The question then arose: Would the cloud provider be able to satisfactorily customize our standby and backup environments?

Conclusions

Developing our own cloud solution proved to be an interesting experience. We gained an appreciation of the difficulties of creating a successful cloud environment. We also became far less comfortable with the idea of placing applications in a computing facility over which we had no control, impact, or ability to tune the application, etc.

Users of cloud vendors must trust that the facility is doing things correctly. This requires very rigorous testing to replicate the production environment in a standalone system when problems arise in order to demonstrate that the cloud is or isn't the cause of the trouble. Cloud users also need to ask many questions about security, access to their information, and personnel of the provider.

Cloud computing sounds like a very useful way to centralize data, remove many of the back-end burdens of system development, and create a worry free user experience. However, this is not an approach to jump into blindly. It is not as simple as allowing someone else to host your application or large virtual computer. This may change in the near future but for now, cloud computing should be treated like all other "silver bullets." There are some applications where using a cloud makes sense and others where it does not. The moral of the story is: "Proceed with caution."