PERFORIC
performance is the key

PERFORIC
performance is the key

WHITE PAPER

Performance testing with cloud computing

and other application performance and porting considerations

Cloud performance

There is a steady stream of applications that are now moving to the cloud. The initial hesitation of handing over sensitive or proprietary data to some cloud outfit "out there" seems to be waning. The presence of big name companies in the space and improved security has helped the cause. The big advantage to cloud computing over traditional hosting is ease and flexibility. In the traditional model you would need to call a hosting company, have them set up servers per your need and gain access to the environment. This was not always quick or painless. You were stuck with a fixed capacity and as your demand increased you had to go through the whole process again to add new hardware to your pool. With cloud computing, all this can be done in a few clicks and you pay only for the amount of computing power you need.

The ease of use and flexibility of cloud computing is particularly well suited for performance testing

The ease of use and flexibility of cloud computing is particularly well suited for performance testing. The dilemma of sizing an environment for performance is to make sure it has enough capacity to handle peak loads but no wasted capacity. Both the load driver environment and the actual deployment environment need to be optimized. The cloud makes this process easier and as a result, your hardware allocation more cost-effective.

Now lets separate the load environment from the deployment environment and see how the cloud can help in performance testing. The load environment is the hardware needed for the load driver software to push load on the test system. There are various tools for simulating user load and their hardware demand can be substantial when pushing loads of thousands or hundreds of thousands of users. The deployment environment is where you have deployed your product, either for testing or production. This is the environment against which the load will be run. If you are looking for saturation testing, that is looking at the pure limits of your software, then you will want to eliminate the internet from the equation and have both the load and deployment environment on the same cloud and on the same network. Saturation testing is useful in the early stages of performance testing to identify system issues. Even if you have no intention of deploying to a cloud due to data or security concerns, you can leverage the cloud to test your application. You can use seed or sample data when you deploy your app to the cloud if you have concerns on data privacy.

This can be a huge cost saving for a company. A dedicated performance lab requires expensive hardware that can sit idle in the early stages of product development. The flexibility of the cloud means you only pay for the resource when you need it. Of course, if you are planning on deploying your application to a cloud, then it's a no-brainer to use the cloud for your load environment as well. You won't have to do any extrapolation of the numbers you get on the cloud to your actual production environment.

If you don't want to deploy your application to the cloud, you can still use the cloud as your load environment. In this case, you will want to be careful in choosing a cloud service as you will want it to be geographically close to your deployment environment. Large cloud providers are making this easier by having locations spread around the world. Performance testing from your cloud load environment to your local deployment environment can still be tricky due to the presence of the Internet in between. It's simply too much of an unknown and network latency can cause your load to be reduced by the time it gets to its target. Some people argue that this kind of testing is more real world and that is a valid point. However, the goal of performance testing should be to remove the unknowns from the equation so you can have a consistent load on your system and going over the Internet can hamper that. The best model for performance testing is to have both load and deployment environments on the same network.

The best model for performance testing is to have both load and deployment environments on the same network

Lastly, the cloud model of pay as you go seems to be catching on with some of the load driver vendors. This can result in substantial savings as some of the load test tools tend to be expensive with their fixed licenses and a pay as you go model can be cheaper, especially for smaller workloads. Some load drivers are already certified and available on clouds with this new license model.

Even with all the hype surrounding the cloud, there are some real benefits to be had in using it while doing enterprise performance testing.

Benefits of Operating-system level virtualization

Virtualization is a broad term and usually denotes a way to remove the hardware dependency from software. Operating-system level virtualization is a means by which multiple operating systems can exist on the same hardware. There was a time prior to this type of virtualization where each type of hardware came with its own proprietary operating system.

Windows could only run on x86 systems, AIX could only run on IBM Power Processor systems, Solaris could only run on SPARC systems etc. With the advent of virtualization, different operating systems can run on different hardware. This led to a rush to the lowest cost hardware with virtualization providing a way to mimic any operating system. There are a lot of vendors providing virtualization for the cheap x86 chipset.

The early knock against virtualization was performance. Typically, the native operating system provided much better performance than the virtual operating system. This was understandable as the hypervisor or the virtualization layer was not optimized for the native architecture. Hypervisor technology has now improved to the point that on some platforms the virtualization performance is similar to native performance. A native hypervisor that runs on bare metal, sometimes called a Type 1 hypervisor, can provide very good performance (as compared to a Type 2 or hosted hypervisor that runs on a conventional operating system). Today the cost benefits of using virtualization with cheaper hardware typically outweigh any performance drawbacks.

Today the cost benefits of using virtualization with cheaper hardware typically outweigh any performance drawbacks

Virtualization provides a way to bundle an operating system as an image containing all the data needed to recreate that same state of the operating system on a different machine or different hardware. This is a huge benefit for QA and for performance testing. A system could be tuned for performance on one machine and then deployed to multiple machines via virtualized images. All the settings like application server settings, JVM settings, database settings and operating system level settings are easily preserved and deployed across a multi-server environment in this way. This reduces the risk of error prone manual tuning across different servers and reduces the time taken to deploy in a complex environment (deploying a virtualized image is a proven and efficient process similar to a file copy).

Cloud computing makes use of virtualization to provide cost benefits and ease of use when making large data storage available. If you are testing or deploying in a cloud and are already using operating-system level virtualization in-house, it’s an easy step to move your image to the cloud. Your software can be certified for a certain throughput (like users per second with sub second response times or all SQL queries returning sub-second query times) on one image and then deployed to a larger environment in the cloud to see how it scales.

Since almost no software is infinitely scalable due to some bottleneck showing up at one level or the other, you can certify your solution to a certain threshold with a given number of images on a certain type and number of CPUs. This certification can happen in a cloud or in-house but the point is that virtualized images allow a concrete unit to be defined in terms of performance and scale that unit out in a predictable manner.

Checklists eliminate the need to remember things and allows the experts to focus on one thing at a time

Use of checklists during software deployment

Checklists are an invaluable tool in many fields that require precision and that have little tolerance for error. Pilots use checklists before takeoff to make sure all systems are functional. Doctors use checklists during surgery to make sure they don't miss anything obvious like leaving sponges or other medical instruments inside the patient after surgery. There has been a lot of research done to show that in high pressure situations like cockpits and operating rooms, checklists not only help avoid obvious errors but keep the key actors calm. This is because it eliminates the need to remember things and allows the experts to focus on one thing at a time.

Anyone who has been in the deployment/go-live stages of a software project would liken it to a high-pressure environment similar to cockpits or operating rooms. Unfortunately, not all companies use a similar checklist approach to ease this process. I have seen countless times where the logging level on an application server is turned up too high and causes wait issues or a database job is scheduled for peak times causing the application to become unresponsive. In a production environment, these issues are difficult to debug and time pressures on getting the site functional again only increases the misery for the support staff involved.

A software or product deployment checklist will be different based on the technology involved so it's very difficult to come up with a generic checklist that applies to any and all situations. There should be a checklist for each of the major physical components in the system like the application server, web server, network, firewall and database. There should also be checklists for the logical components in the system like the security infrastructure or third -party integration components. The key to a good checklist is that it should have enough detail that someone can follow it literally without thinking. I have provided some high level things to look for above but it is up to a particular software shop to fill in the details based on its products and technology. For example, hardening a database for production is a well-known task for each database vendor and should be part of the database checklist. Most application server vendors will provide details on tuning an application server for production and these steps should be part of the application server checklist. These checklists should be constructed jointly by the development team that will have the details of the software product settings and the services team that will have real world experience from deploying at customer sites.

Performance testing is something that should be part of any deployment checklist

Performance testing is something that should be part of any deployment checklist. A scaled down environment should be used to run load against and tune for optimal performance. This is easier said than done in a complex environment where there might be call-outs to other live systems and certain functionality that can only be made available to the production system. Techniques like stubbing out calls to external systems can be used to make the scaled down environment functional and self-sufficient but they can also hide issues lurking when the actual calls are made. The closer you can get to making the scaled down system similar to the actual production environment, the better off you are. However, just because you cannot get them to be exactly the same does not mean that performance testing in a smaller environment is a waste of time. A lot of issues can be uncovered on the local system itself. Once tuned, the trick is to get the exact same settings and images to the production system. Virtualized images are one way of doing this. Again, the key is to make sure that the production system software and settings are exactly the ones that are used in the performance test system.

The use of exhaustive checklists that include performance testing the production environment, even if in a scaled down manner, should greatly reduce any surprises when going live.

Summary

One of the biggest concerns for any business and IT organization is uptime. Application performance can impact revenues, customer satisfaction, employee productivity, data center efficiency and software and hardware licensing costs. There are several ways to manage application performance including performance testing, benchmarking, capacity planning and performance monitoring. These services are available both on-premise and as hosted service offerings and can help you ensure the health and the performance of your enterprise applications both before you launch and while in production. Organizations of all sizes should consider these services when application uptime is critical to business productivity.

About Perforic

Perforic provides flexible, high quality and cost-effective enterprise application porting and performance testing services for software product companies and internal development organizations. Our team consists of seasoned veterans with vast experience in performance and porting issues. We have worked with companies of all sizes – from startups to Fortune 500 companies - and have deep expertise in J2EE, .NET and database technologies in highly scalable, three-tier enterprise application environments. The company is based in Cambridge, MA and has 24 x 7 operations with locations and resources in the US and Asia. For more information, please visit http://www.perforic.com/index.html.

Perforic LLC
955 Massachusetts Avenue, #309
Cambridge, MA 02139
info@perforic.com