WHITE PAPER

Performance testing open source software

and other application performance and porting considerations

Open source and software performance

It is now entirely possible to run an enterprise application on all open source software

The arrival of open source software has ushered in a new era of democratic software development. We are no longer beholden to large corporations that provide expensive proprietary software packages with even more expensive support terms. Open source packages are usually free and are worked on by a legion of independent software developers who are constantly making improvements to the software. Open source software foundations like Apache provide a loose framework that allows disparate developers to make changes to software hosted on a source control system managed by the foundation. The foundation provides the infrastructure by which the software is packaged and made available to the public. The source code for open source packages is available for all to see. This mechanism has led to an explosion in open source software from web servers to application servers and databases. It is now entirely possible to run an enterprise application on all open source software. The only item needing considerable capital outlay is the hardware and with hardware costs coming down every year, even that is not as expensive a proposition as it used to be.

The flip side of all this choice with open source software is the issue of stability and performance. This is where large companies with their proprietary packages have an edge. They guarantee the thing most important for companies when they are buying mission critical software: stability and performance. Large companies can do this because they tightly control their software environment and have extensive testing methodologies. Open source packages by contrast, are atomic packages that have been tested individually but not together as an enterprise solution. This is no fault of the open source developers. They are putting together a solution to meet a specific need.

It is analogous to a wheel manufacturer for a car. They can test their wheels to make sure it meets all safety standards but if the wrong size wheel is put on a car, it can jeopardize the handling safety of that automobile. The car needs to be tested as a whole. Similarly, once different open source packages are assembled to create a final solution, that complete solution needs to be tested. This is where companies who have extensive experience with open source packages AND testing comes in. Ideally, these companies have their own labs that can certify the final solution.

Once different open source packages are assembled to create a final solution, that complete solution needs to be tested

The other issue to be aware of with open source is the issue of upgrades. Typically, proprietary packages are upgraded all at once and the company providing the package makes it easy for their customer to upgrade. Since open source software has individual components from different sources, upgrades are trickier. An upgraded version of one piece of the software might not work with the other components. There might be performance issues introduced as part of the upgrade. It doesn't help that with open source, usually there is no one company to turn to in case of problems. One's best bet here is to look for a company that can certify an open source solution. This company can serve the same role as a large multi-national company supporting its proprietary solution, but at a comparatively lower cost. As with any company you put your mission critical infrastructure in the hands of, you need to ensure that they can support your critical assets 24x7.

The do's and dont's of outsourcing

This article applies not just to performance and portability testing companies but also to any software company that either currently does or is interested in doing outsourcing. Outsourcing has become a political hot potato of late but I am not interested in the political aspects of the debate. I am coming at it as a software professional that has worked with outsourced teams for the last twelve years and who is only interested in the productivity aspect of the equation.

In the early days, outsourcing was looked on merely as cost savings. The work that the local worker either found boring or tedious was shipped off overseas where workers were only too willing to take on any type of technical work. The local worker, presumably being paid more than the outsourced worker, was used to provide high value work, i.e., either more intellectually challenging work or that requiring customer interaction. Over the years, the quality of outsourced talent has improved and now there are even outsourced jobs that cannot be filled in the US due to a lack of that specific talent. The use of teleconferencing and web conferencing has reduced the distance between onshore and offshore teams to the point that they can now be looked at as complementing each other. As the saying goes, you get what you pay for, so those now looking to outsource only as a means to save money will not be very successful, either with the quality of the work or with morale issues onshore due to the stress of having to work with a challenging offshore team.

The new mantra for outsourcing needs to be productivity

The new mantra for outsourcing needs to be productivity. Countries in Asia have a 10 – 12 hour time difference which means that work can go on 24x7 between onshore and offshore teams. The use of centralized servers and remote access software like VNC or Microsoft Terminal Server means that teams can be working on the same machines together. When one team signs off, the other team can pick up exactly where they left off. During software development, coding of different modules can be distributed onshore and offshore with regular integration checkpoints. Good source control software like Clearcase allows seamless checkins across different sites with synchronization built in. Use of web conferencing, teleconferencing and instant messaging allow disparate teams to constantly be in communication. The use of clear functional documentation allows teams to be on the same page while developing code. Good process around unit testing and nightly builds helps catch issues sooner (this applies to onshore as well as offshore but is critical when managing distributed teams).

The gauge of how effective an outsourcing team is should be based on if the company is getting to market quicker and implementing features quicker rather than how much they are saving

Using your offshore team as an extension to your existing team will allow both teams to be more effective. Good communication with the offshore team will allow them to feel plugged in to the onshore team and will produce benefits that knowledge workers produce when they are engaged and active in the software development process. The gauge of how effective an outsourcing team is should be based on if the company is getting to market quicker and implementing features quicker rather than how much they are saving. An ill-managed or inefficient offshore team can quickly cost more in delays and quality issues than the cost savings they are supposed to be generating.

Here are some basic do's and dont's :

- Do have a well-defined software development life cycle
- Do communicate requirements clearly across teams
- Do have the discipline to write functional specifications and requirement documents rather than just verbally explaining them to onshore developers
- Do have a central set of servers with remote access for both onshore and offshore engineers. This is more important for QA and performance testing than software development
- Do have a centralized source control system that can synchronize between different sites
- Do invest in communication software like web conferencing and instant messaging
- Don't look at outsourcing as simply a cost saving mechanism. Rather look at it as a 24x7 operation between onshore and offshore teams
- Don't engage with offshore teams without checking references and without a visit to their facilities
- Don't expect offshore teams to pick up all the graveyard shifts and make all the conference calls at midnight their time. These should be equally split between onshore and offshore so that no one team feels second class. This is especially important as more and more offshore talent produces the same level of work as the onshore workers and to keep talented workers on your team, flexibility is required

Mobile performance testing

Web sites handle mobile traffic through either implementing the Wireless Access Protocol (WAP) or modified HTTP

There are more and more mobile devices coming on the market today providing greater and greater functionality. With the advent of the iPhone and Blackberry as the dominant players in the PDA market, consumers expect to be able to access all their data from their mobile device. Apple has created the hugely successful iPhone App Store and RIM is getting into the action with similar applications for the Blackberry. This has put a huge strain on the underlying infrastructure that is actually serving up the data. Enterprise applications today not only need to serve up data to the traditional browser on a laptop or desktop but need to serve up data to mobile devices. As mobile devices proliferate, so does the strain on the enterprise servers serving up this data. People no longer need to be sitting in front of a computer to access data or their favorite sites. They can now do so from a mobile device, so they access these sites more often and stay on them longer.

Web sites handle mobile traffic through either implementing the Wireless Access Protocol (WAP) or modified HTTP. In either case, since wireless networks don't have the same speed as broadband, they try to reduce heavy graphics and other data intensive themes when serving data to a mobile device. This explains why the look and feel of the same web page is sometimes different when viewed through a mobile device vs a traditional computer. It is important to note that any transformation done for a mobile device is done typically at the web server tier so the underlying tiers still need to do the same amount of work to serve up the data. This includes the application servers and the database. WAP or HTTP is a layer on top of the traditional web site infrastructure that is mostly concerned with mobile data presentation.

The way to do performance testing on mobile devices is essentially to test the underlying tiers to make sure they can handle the load. This can be done using traditional load testing tools like LoadRunner. Since most web sites provide both mobile and web content served by the same underlying mechanism, the load-testing tool can be used against the web presentation interface. This would also ensure stability for the mobile interface. Once the stability of the underlying tiers has been established, we need to turn to the web tier or the tier that is handling the data transformation for the mobile device. As we are focused on performance testing here, we will assume that this tier is functionally tested and producing the data for the mobile device the way it is meant to. In order to test how the WAP tier will hold up under load, we can use an emulator or an actual device. There are pros and cons to either approach. Emulators are cheaper than getting a large number of actual devices but on the other hand they might have issues of their own that get in the way of testing for performance. Actual devices are expensive and can be difficult in terms of coordinating a large number of users to simulate a load test also known as a blitz. The best solution depends on each use case and customer. As has been noted earlier, wireless networks typically are much slower than broadband, so a performance test using an emulator on a broadband network could show much better performance than when running the test on the actual mobile device. If the pages are too data intensive, the performance issue could show up only for the mobile device. The good news is that there is no need to do a load test to detect this situation. A functional test on a single device should uncover any bandwidth related issues.

Since most web sites provide both mobile and web content served by the same underlying mechanism, the load-testing tool can be used against the web presentation interface

Mobile performance testing consists of two parts: testing the underlying tiers for stability which is no different from traditional web site performance testing, and testing the mobile interface which poses more challenges since it's a relatively new area.

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
performance is the key

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