

Leveraging the Cloud and Open Source Technologies in Performance Testing:

By Rajini Padmanaban, Director Testing Engagements, [QA InfoTech](#)

With the current day trend where customers' product expectations are rapidly changing and everyone needs products that are really rich, fast and intuitive to use, the kinds of testing that are imperative are very different than a few years ago. With excellent internet bandwidth that is now available, products and applications are all scaling very well in their performance and this is something the customers are also expecting from the products they use. That said, such a fast and flawless implementation of products is not as easy as it sounds. There are a lot of challenges associated with meeting the product's performance goals which include:

1. Increasing complexity of product architecture
2. Several interfaces that the product has with external systems and dependencies
3. Increased user base, leading to heavy load on the system
4. Performance goals undergoing a lot of change owing to competition in the market and demanding customers
5. Faster time to market for product release, which shrinks available time for performance testing
6. Shortage in availability of specialized testers for performance testing
7. Investment in performance testing infrastructure and tools, which are often very expensive and may become obsolete soon

If you see the above list, some of them are product and user specific challenges while some are test process challenges. The key to solving the product and user specific challenges is really around developing a robust performance test strategy and adequately testing the product for its performance before release. So, what we really need to be looking at is how to address the performance testing challenges; this forms the core of our discussion below.

Performance testing has long been in existence, however it has rightfully been given a lot of thought around specialization, domain centric testing, using the right infrastructure which mimics end user environment etc. only in the recent years. As product development technology advances with use of secured protocols, rich internet applications, service oriented architecture, web services etc. developing performance testing scripts has also become more complex. No longer will a regular "record and play" tool work across a major set of products. There are a lot of very advanced performance testing tools that are entering the market from commercial players like HP, Microsoft, IBM which are constantly being enhanced to keep pace with the changing product technology. However, taking advantage of such tools does not come for free. These are all very expensive Commercial of the Shelf (COTS) tools, which you need to

very carefully evaluate before investing. These might especially make sense for a product company to invest in, but not so much for a testing services company, because the services company is often required to use a tool that its clients have aligned their product with. On the other end of the spectrum, we have some excellent open source testing tools available for performance testing lately, which give the COTS a run for their money. A lot of companies, including product companies have started looking at such tools as good alternatives from a cost and feature standpoint, to use in their performance testing efforts. Jmeter is one such tool which a lot of companies are leveraging in the recent years. Again talking of trade-offs here, Jmeter is an open source tool available free of cost, easy for a tester to ramp up on, but is not comprehensive in its feature set, unlike the COTS. It has quite a few limitations around its reporting capabilities, which is a very important feature in performance testing, especially when the tester is handling large volumes of data. Also, this is data that the management and business teams are going to be interested in viewing, so a performance testing tool has to certainly offer very rich reporting functions.

This is where a company can draw its balance and decide to go with using an open source tool to take advantage of what it has to offer, but also invest in building reusable intellectual property (IP) on top of it, to address the tool's limitations. Doing so, is beneficial for all entities involved as seen below:

1. **Clients (Product companies)** are able to get performance testing done at a cheaper cost without compromising on quality
2. **Vendors (Test Service companies)** are able to offer good value add services to their clients, differentiate themselves in the services market amongst their competition and also offer interesting out-of-project opportunities for their employees to work on building such core IP
3. Very challenging and interesting opportunities and thus good career path progression avenues are available for the **employees of test services companies**

Such reusable IP and frameworks often also serve as productivity enhancers – e.g. data generation, especially user data creation is quite an overhead in a performance testing effort compared to regular functional testing. If the performance testing tool that you choose does not offer good user data generation features out of the box, this is an area worth investing some time on upfront as it will save time both in the current test cycle as well as future cycles. Automating such time consuming and monotonous jobs, also frees up time for the tester to focus on more important areas such as performance benchmarking, system analysis, competitive data analysis etc. Also, to maximize the performance testing efforts, the product company should involve the performance testing team right in the early stages of product design. This will help the performance test team work in unison with the business team to understand end user's performance

expectations from the product under development; Understanding this will help the team chalk out “measurable” performance goals for the product and help the entire product development team understand these goals on the same page. Another benefit to such early involvement is the suggestions that the performance test team can provide on the product architecture from a performance angle (e.g. it could be the number of interfaces the product has, how certain web service calls are made, threshold values that are being set, production system configuration parameters etc.) which will help reduce the number of performance bugs in the test cycle. Such interactions between the developer and tester truly help build a quality product in the limited available release time.

We talked about the challenges around performance testing tools and availability of time to test for performance and some potential solutions to address them. Another major challenge we broached above was the availability of the right test infrastructure on which performance testing can be done.

Performance testing is one area which can be quite time consuming, especially in doing tasks that aid the core testing activity itself. For e.g. setting up the test infrastructure which mimics the production deployment environment, user data generation, report generation etc. It is considered a best practice, if the product or services company can find the right and optimal solution to reduce the tester’s efforts on these supplementary tasks; this will then give the tester more time to focus on tasks that really need his attention such as deciding on the performance test strategy, scenario identification, scripting, test execution, result analysis, benchmarking and regression. That brings our discussion to how we can help the tester spend less time on setting up the test infrastructure – in the more recent years, the answer to this question is using the “Cloud”. The cloud basically offers services to the end users on a pay per use or lease model. Depending on what the service is, a cloud could basically offer “Software as a Service (SaaS)”, “Platform as a Service (PaaS)” or “Infrastructure as a Service (IaaS)”. We are specifically herein interested in “**Infrastructure as a Service**”, where one connects to machines on the cloud and uses them for the performance testing effort. Doing so, offers a lot of benefits to the product or services company, which include:

1. **No locking of funds in expensive hardware for performance testing** – this is a very compelling benefit since performance testing often needs very high end hardware with several core CPUs, lots of RAM etc.
2. **No risk of hardware becoming obsolete** – Besides just the initial investment, one of the concerns of owning the hardware is that hardware specifications are changing by the day. So, investing in such expensive hardware just for performance testing will not yield the required “return on investment” for the product or service company.

3. **Good control on overall test costs** – since you pay for the test hardware on the cloud only on a “pay per use model” or “possibly a minimal seat cost” model, the costs associated with test infrastructure are kept low and under control. Costs are kept under control also from the standpoint that the tester’s time that is required to setup the machines is now very minimal or non-existent. Such savings in time also translates to savings in costs.
4. **Improved accuracy in testing results and better interaction in the product teams** – Performance testing is often an area where reproducing bugs/issues is very difficult especially when global teams are involved. This is often because the entire team does not have access to the same machine setup and setup is a huge dependency factor in getting predictable results. However, when the team uses infrastructure on the cloud, anyone on the team can access the same setup, improving the chances of reproducing a given issue.

A lot of commercial clouds are available these days at very competitive prices and feature sets for you to choose from. Some of the more popular ones are from Amazon, Google, Microsoft.

Before deciding to go with a cloud based solution for IaaS, a product or a services company should clearly plan for its cloud usage strategy. Some of the core things to consider here include:

i) Clear cut test strategy:

Identify testing objectives – cost benefits, scalability, and return on investment

Identify types of Performance testing to be conducted

ii) Infrastructure - Identify the following:

Hardware and software requirements

Test automation tools

Number of concurrent users

Network bandwidth

Memory requirements

Storage requirements

Application usage (Frequency)

iii) Service providers:

Must have high QoS

Must be able to provide the entire suite of end to end services – infrastructure, software licenses, setting up and dismantling the environment

Must take minimal time to set up and dismantle the infrastructure

Must have proven record of adhering to SLAs

When all of these are carefully planned and a cloud based test infrastructure setup is leveraged, it brings in a lot of time and cost savings to the service provider and product company, helping bring down the overall product development costs.

One of the last core challenges of performance testing we mentioned above is the lack of performance testing engineers. Performance testing is a highly specialized area and a very important area that plays a critical role in a product's acceptance in the market. Every company has to pay close attention to how to build a center of excellence around performance testing that defines associated test processes, tools, best practices and also how to build, groom and nurture performance test experts. This includes hiring the right talent, spotting the right talent within the company that can be trained on performance testing, offering periodic workshops and sending the experts to the right trainings to further build their skills and keep them motivated, investing in the right resources (infrastructure, tools, internal projects such as building internal IP) that will empower these engineers to give their best on the job and also motivate them. All of these go a long way in giving the company a leading edge in building the right performance test team and performance test center of excellence, helping them meet the overall performance goals of the product under development.