By Alexander Podelko

Performance Testing in the Cloud: Look Beyond the Word

The best combination of options for you depends on the goals of performance testing. Performance testing in the cloud (or from the cloud) makes sense for certain types of performance testing. For example, it should work very well if we want to test how many users the system supports, whether it would it crash under a load of X users, or the number of servers we need to support Y users, but only if we are not too concerned with exact numbers or variability of results (or even want to see some real-life variability).

Even in this case, the assumptions are that we aren't introducing any bottleneck using the cloud (for example, saturating network bandwidth between the load generators and the system under test) and leave it to the cloud provider to worry about whether our tests will impact other cloud tenants.

However, it doesn't work well for performance optimization, when we make a change in the system and want to see how it impacts performance. Testing in a cloud with other tenants intrinsically has some results variability, to the extent that we don't control other activities in the cloud and, in most cases, don't even know the exact hardware configuration. For example, if the system scales out by automatically creating an additional application instance, the new instance may be outside of the network segment where other servers are. The effects may be even more sophisticated in case of Platform as a Service (PaaS) or SaaS clouds. So, when we talk about performance optimization, we may still need an isolated environment.

One interesting case is when the system is created to be used in a cloud, which probably will become more and more common over time. The first thought would be that it simplifies the choice – you just test it in the cloud where it is supposed to be deployed. Still, it will not work too well if you need to do performance optimization or troubleshooting and want tests to be completely reproducible. In this case, you may need something like an isolated private cloud with hardware and software infrastructure similar to the target cloud, and monitoring access to the underlying hardware to see how the system maps to hardware resources and if it works as expected. Real-world network emulators may be used to make sure that performance testing is representative of how the system would be used in production – otherwise we would not be taking into account such factors as network latency, bandwidth, jitter, etc. So, if we need optimization for cloud software, we may still need a lab – but the lab should be more sophisticated to emulate the cloud environment and real-world network conditions. An ultimate example of such a lab is the lab Microsoft created for testing IE, described at *blogs.msdn.com/b/b8/ archive/2012/02/16/internet-explorer-performance-lab-reliably-measuring-browser-performance.aspx*.

The cloud introduced new opportunities and challenges to performance testing, but specific pros and cons vary significantly depending on your environment and goals. The term cloud is overused and covers a lot of different options. If we want to understand how the cloud may impact performance testing, we should consider all these options separately as they result in completely different performance testing contexts.

In performance testing we have two main components: the system under test and load generators. There may be other components for monitoring, results analysis, etc., but they are not so important in the context of this discussion.

When we talk about load generators, we have three main options:

- Locally, the traditional option (for example, in a test lab).
- As a service. This option has existed for a long time (for example, load testing services provided by Gomez, Keynote, and other companies). While we can refer to it as a SaaS (Software as a Service) cloud now, the only real change is that we now have more such companies (and, respectively, more choices) because it is easier to start such a service using cloud to provide the infrastructure.
- In IaaS (Infrastructure as a Service) clouds. This is a new option which makes it easy to access a large number of remote load generators. It was always possible to have a load generator on a remote machine, but now it is much easier. Some tools provide help with cloud deployments, which may be very handy when you need a large number of load generators for a large-scale test.

When we talk about the system under test, in addition to having the system locally (which may be anything from a development machine to the production system) we may now deploy it in a cloud. This helps to overcome one of the main reasons of not testing full-scale setups – lack of hardware resources. Now you can access as much resource as you like when you are ready for it. However, it may not be exactly the same kind of hardware and software that you use in your production system, so in getting closer to the scale of the system you may be farther away in terms of the details of the environment.

Thus, we have different options for the system and load generators deployments, and which option (or combination of options) would be the best depends on the goals of performance testing. For example, some typical performance testing scenarios might be:

- System validation for high load. Outside load (service or cloud) against the production system may be the best option here. We have a wider scope of testing, but lower repeatability.

- Performance optimization/troubleshooting. An isolated environment may be the best option here. We have a limited scope, but high repeatability.

- Testing in cloud. This may be the best option for periodic tests to lower costs. We have a limited scope and low repeatability.

So, factoring the cloud into performance testing, we have at least two major alternatives (with a variety of subtler options): coarse performance testing in or from the cloud with inherent variability (and probably some savings on hardware and configuration costs) or granular performance testing and optimization in a isolated environment (thus avoiding variability with, probably, higher hardware and configuration costs). For comprehensive performance testing you may even need both lab testing (with reproducible results for performance optimization) and realistic outside testing from around the globe (to check real-life issues that you cannot simulate in the lab). Doing both would be expensive and makes sense only when performance really matters – but if you are not there yet, you may get there eventually. ∎

> about the author

For the last fifteen years, **Alex Podelko** has worked as a performance engineer and architect for several companies. Currently he is Consulting Member of Technical Staff at Oracle, responsible for performance testing and optimization of Hyperion products. He blogs at *www.alexanderpodelko.com/blog* and can be found on Twitter as *@apodelko*.